1995年6月1日発行（毎月1回1日発行）第14巻6号通巻158号 昭和58年11月2日第三種郵便物認可

PERSONAL COMPUTER MAGAZINE for MZ, X1, and X68000

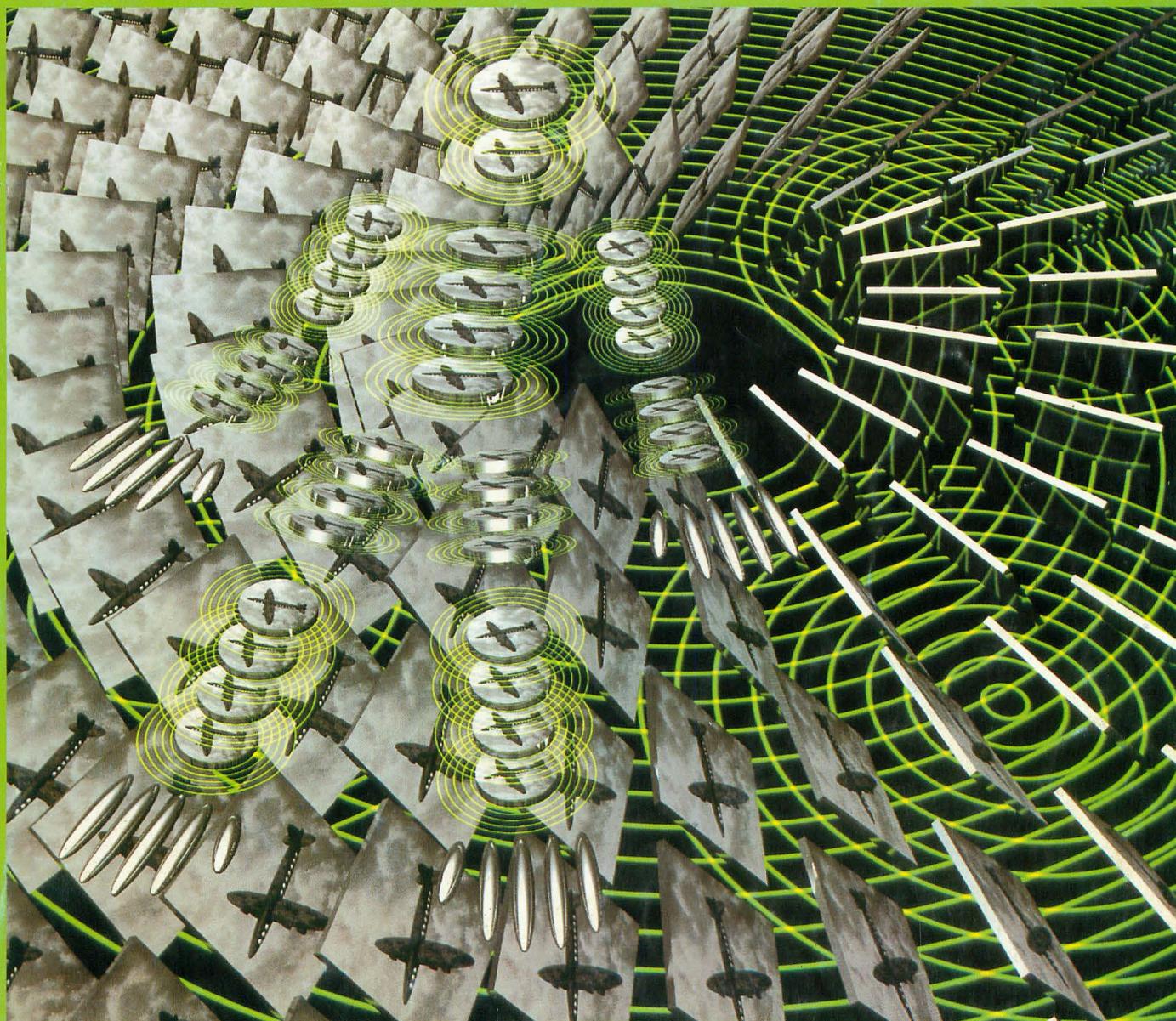
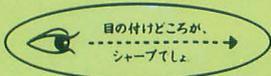
特集 Open the SX-WINDOW

特別企画 X68000周辺機器パワーアップ計画

第6回アンケート分析大会

新製品紹介 Xellent30s/学研統合電子辞書 for SX-Window

6
1995

SOFT BANK オー!エックス
定価680円

■実画面：1,024×1,024ドット、表示画：768×：512ドット

●画面は広告用に作成した、機能を説明するためのイメージ画面です。また、各種アイコンなどは、SX-WINDOW ver.3.1がもつ機能を使って作成したもので、標準装備のものとは異なるものもあります。
●本広告中の「シャーペン」で表示している文字のフォントはツァイト社の、「書体倶楽部」のフォントを使用しています。

❶「パターンエディタ」で作成したデータを背景に設定可能。

❷日本語フロントプロセッサ ASK68K ver.3.0の辞書メンテナンスがウィンドウ上で可能。

❸ESC/Page、LIPSIII、PostScriptに対応したプリンタが利用できます。

❹付属アプリケーション「シャーペン」編集例。文字ごとに文字種・文字の大きさの指定、装飾が可能。またインライン入力をサポート、イメージデータの貼りつけもOK。

❺512×512ドットの範囲内で65,536色の表示が可能。

❻「CGAウィンドウ」、65,536色（最大）のコンピュータアニメーション表示が可能。

❼異なる画像フォーマットへのコンバートが可能。

❽アイコンデータや背景データを作成する「パターンエディタ」。

❾オリジナルに作成したアイコンパターンの例。

❿Human68kやX-BASICのコマンドをSX-WINDOWアプリケーションと同時にタイムシェアリングで実行できます。

■「SX-WINDOW ver3.1システムキット」のお問い合わせは…シャープ㈱電子機器事業本部（液映）システム機器推進プロジェクトチーム　〒162 東京都新宿区市谷八幡町8番地 ☎(03)3260-1161（大代表）へ

フィールドが、膨らむ。

先が、ますます面白くなる。

●

未来への確かなビジョンをベースに
発展性のあるプラットホームとしてのウィンドウ環境を提供する
国産オリジナルウィンドウシステムSX-WINDOW。

●

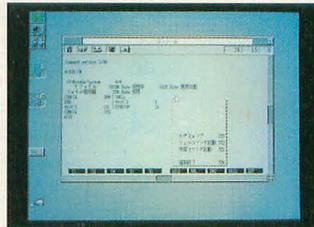
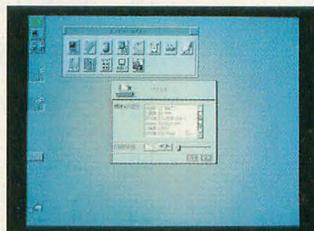
GUI環境や操作環境、高速化へのゆるぎない探求、
マルチメディアの統合的なハンドリング。

●

いま、より多彩なフィールドへ
そのインテリジェンスが展開を始める。

●

次のステージが見えてくる。

●インライン入力のサポート：ASK68K Ver.3.0を利用したインライン入力をSX-WINDOWで実行可能。またシャーペン.Xをワープロとして利用できるよう、さまざまな機能が付加されています。

●コンソールをサポート：Human68kやX-BASICのコマンドをSX-WINDOWアプリケーションと同時にタイムシェアリングで実行できます。
（グラフィックを利用したものなど、SX-WINDOWと処理が重複するものは実行できません。）

●多彩なプリンタに対応：さまざまなSX-WINDOWアプリケーションで利用できるページプリンタドライバを標準装備。ESC/Page、LIPS III、PostScriptに対応したプリンタが利用できます。

SX-WINDOW ver.3.1の
データ利用環境

今も、先も楽しめる。
いつも新展開の予感、SX-WINDOWのニューバージョン。

SX-WINDOW ver.3.1

「SX-WINDOW ver3.1システムキット」CZ-296SS（130mmFD）/CZ-296SSC（90mmFD） 標準価格22,800円（税別）

68買ったら EXEクラブへ入ろう!

EXEクラブって何だ?

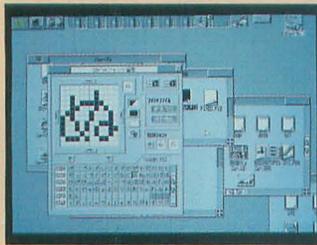
X68030/X68000を手に入れて、いろいろチャレンジしたい皆さん。情報のチャンネルは多いほどいいですよね。ということで、EXEクラブは68ユーザーのための水先案内人。あなたのチャレンジを強力にバックアップしますよ。

本体同梱の入会申込ハガキを送るだけで、自動的に無料入会。さらに下記の特典付き。

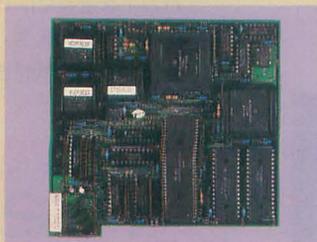
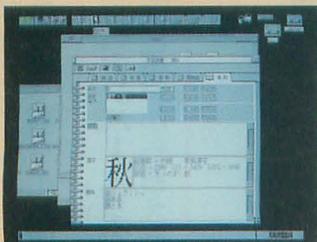
メリット1 会員ナンバー入りのオリジナル会員電卓がもらえる。

メリット2 各種フェアご優待・イベント案内等、数々の特典がある。

Oh!X

CONTENTS

特集　Open the SX-WINDOW

特別企画　X68000周辺機器パワーアップ計画

マイコンショウ'95

Xellent30s

学研統合電子辞書 for SX-Window

こちらシステムX探偵事務所

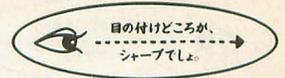
〈スタッフ〉
●編集長／前田　徹　●副編集長／植木章夫　●編集／山田純二　高橋恒行　●協力／有田隆也　中森　章　林　一樹　吉田幸一　華門真人　朝倉祐二　大和　哲　村田敏幸　丹　明彦　三沢和彦　長沢淳博　清瀬栄介　石上達也　柴田　淳　瀧　康史　横内威至　進藤慶到　菊地　功　伊藤雅彦　●カメラ／杉山和美　●イラスト／山田晴久　江口響子　高橋哲史　川原由唯　●アートディレクター／島村勝頼　●レイアウト／元木昌子　加藤真二　●校正／グループごじら

表紙絵：塚田　哲也

E　N　T　S

1995 JUN.
6

■広告目次

ビデオグラフィックスの
世界へ。
■お問い合わせは… シャープ株式会社
電子機器事業本部システム機器営業部 〒545 大阪市阿倍野区長池町22番22号 ☎(06)621-1221(大代表)
X68030

1,677万色対応、ビデオ映像を高画質・高速取り込み

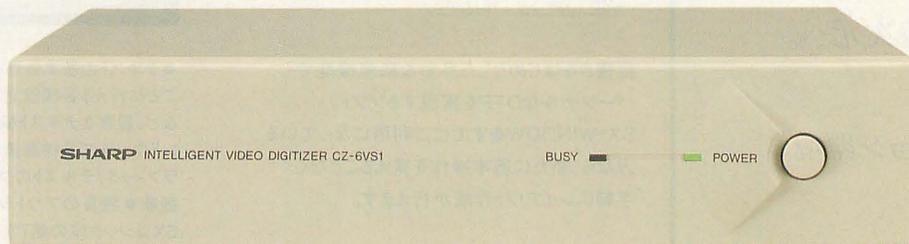
テレビやビデオ、ビデオディスクなどの映像をX68シリーズやMacシリーズ※1の動画・静止画データとして高速取り込みが可能、いわば“ビデオスキャナ”とでも呼びたいビデオ入力ユニットです。1,677万色対応、最大640×480ドットの高解像度※2。動画・静止画の手軽なハンドリングが、新たなグラフィックシーンを創造します。

※1 MacintoshはIIシリーズ以降の機種に対応、ディスプレイ解像度が640×480ドットの場合、取り込み可能な範囲は、160×120ドット、320×240ドットのサイズになります。
※2 X68030/X68000シリーズでは、1,677万色はデータ作成のみに対応。表示は最大65,536色、解像度は512×512ドット。また、Macintoshは機種により表示色数が異なります。

アプリケーションツール「ライブスキャン」を標準装備

動画や静止画を簡単に保存できるアプリケーションソフト「ライブスキャン」※を標準装備。取り込んでいる映像を表示したり、残したいシーンを簡単に静止画保存したり、手軽な動画・静止画ハンドリングでパソコンの可能性をさらに広げます。X68030/X68000シリーズ用SX-WINDOW対応版とMacintoshシリーズ用QuickTime対応版の2種類を同梱しています。

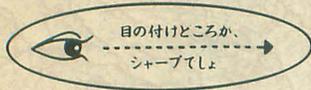
※SX-WINDOW版はバージョン3.0以降（メモリー4MB以上）、QuickTime版はMacintosh漢字Talk7リリース7.1以上のシステムとQuickTime1.5以上（メモリー8MB以上）が必要です。

1,677万色対応の高速映像取り込み、動画・静止画の手軽なハンドリングが、新たなマルチメディアシーンを創造する。

■SCSIインターフェイス採用：パソコンの専用I/Oスロットを使わずに接続可能になり、汎用化を実現しました。またSCSI-2（FAST）インターフェイスの採用により、データ転送速度の高速化を図っています。X68030/X68000シリーズでは、SCSI-2（FAST）対応のハードディスクを接続することにより、パソコン本体を経由しないで、ハードディスクに直接、動画データをテンポラリデータとして記録することが可能です。パソコン本体のハードディスクへは、記録終了後に、テンポラリデータを変換し動画データとして保存できます。

※CZ-600C/601C/611C/602C/612C/652C/662C/603C/613C/653C/663Cに接続する場合は別売のSCSIインターフェイスボードCZ-6BS1ならびにSCSI変換ケーブルCZ-6CS1が必要です。※CZ-604C/623C/634C/644Cに接続する場合は、別売のSCSI変換ケーブルCZ-6CS1が必要です。
※Macintosh Power Bookシリーズに接続する場合は別売のSCSIケーブルなどが必要です。詳しくはMacintosh Power Bookシリーズの取扱説明書をご覧ください。

■高機能MPUを搭載：クロック周波数25MHzの32ビットMPU/MC68EC020を搭載、高速処理やパソコン本体の負担の軽減を実現します。

●MacはMacintoshの略称です。●Macintosh、Macintosh IIは、米国アップルコンピュータ社の登録商標です。●Power Bookは米国アップルコンピュータ社の商標です。●漢字Talk7はアップルコンピュータジャパン社の商標です。●QuickTimeは、米国アップルコンピュータ社の商標です。●価格には、消費税及び配送・設置・付帯工事費、使用済み商品の引き取り費等は含まれておりません。

標準価格178,000円（税別）

For X68030/X68000series

ORIGINAL SOFTWARE COLLECTION

さらに高度な創造次元へ。
ますます成熟する
そのアプリケーション環境。

NEW アプリケーション

●独自のアウトラインフォントを付属

フォント&ロゴ デザインツール 書家万流 SX-68K

CZ-282BWD 標準価格29,800円(税別)

4MB | ver.3.0 | HD 10MB

フォントやロゴを手軽に作成するためのデザインツール。作成したロゴはクリップボードを介し、シャーペンやEGWord SX-68K、XDTP SX-68Kなど他のアプリケーションで利用できます。

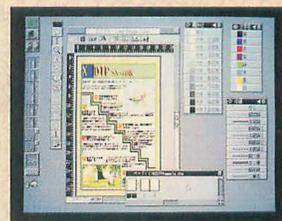
●SX明朝体/SXゴシック体フォント(JIS第1水準&第2水準)を付属●ベジェ曲線のアウトライン編集によるデータ作成●フォントファイル全体にわたってのエフェクト処理●既存のフォントファイルからのデータ抽出、ドローオブジェクトへのエフェクト処理●複数のフォントファイルをリンクして新たなフォントファイルの作成が可能●65,536色表示で確認しながらロゴ作成ができるグラフィックウィンドウ(GRW.X)対応

●パーソナルDTPをX68で

XDTP SX-68K

CZ-291BWD 標準価格35,000円(税別)

4MB | ver.3.0 | HD 5MB

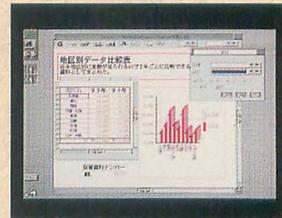
縦書きをはじめとした多彩な編集機能でパーソナルなDTPを実現するソフト。SX-WINDOWをすでにご利用になっている方なら、新たに基本操作を覚えることなく手軽にレイアウト作成が行えます。

●テキストの基本処理をはじめ、テキストフレームごとに行える各種設定、スタイル別の検索/置換など、豊富なテキスト編集機能●グラフィックウィンドウ、そして各種画像フォーマットへの対応●グラフィック/テキストのフレームから独立した罫線機能●独自のアウトラインフォント(SX明朝体、SXゴシック体の第1水準)標準添付●ページの移動/作成/削除がスピーディに行える独立したページウィンドウをサポート

●DTP感覚で自在にレイアウト編集

Datacalc SX-68K

CZ-273BWD 標準価格59,800円(税別)

4MB | ver.3.0 | HD 3MB

SX-WINDOW対応の新世代統合ソフト。表計算、グラフ、データベース、テキスト、罫線の各データを1枚の用紙に重ね合わせ、移動、サイズ変更などDTP感覚でレイアウト編集ができます。

●カルクシートでは、セル番地を意識することのない直感的なセル指定が可能●データベースフィールドでは、同一項目でもデータ型/データ長の異なったデータを管理できるなど、自由な設計が特長●データベースフィールドで入力したデータをカルクシートのデータとして利用したり、カルクシートのデータ変更を自動的にグラフ表示に反映させたり、同一データからさまざまな分析が可能なデータリンクもサポート

SHARP

高速、高解像度。
透過原稿・ADF対応型カラーイメージスキャナ、誕生。
SHARP
JX-330
SHARP IS COLOR

●拡大読み取り時、細かい部分でも忠実に再現。
2400dpi※1やデジタルズーム機能が高品位を守ります。

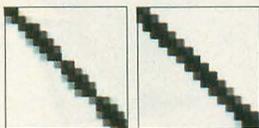
●35ミリフイルムも透過原稿読み取りユニットを使用して読み取り可能。

高速処理を実現。スピーディに作業できます。

A4、300dpiならカラー約13秒※2、モノクロ約1秒※2でこのクラス最高の※3高速読み取りが可能です。大きな画像データを高速転送できるSCSI-Ⅱにも対応。また、最大A4/リーガルサイズ(216.4×355.6mm)までの原稿を読み取りできます。

高解像・高品位。美しさが際立ちます。

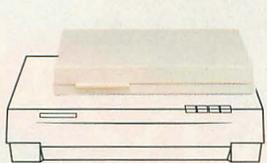
基本解像度600dpi、疑似解像度2400dpi※1の高解像度読み取りで微細な点や線を鮮明に再現します。縮小・拡大は30〜2400dpiの範囲で設定可能です。また、約1677万色で原画に忠実なリアルな色合いを再現します。

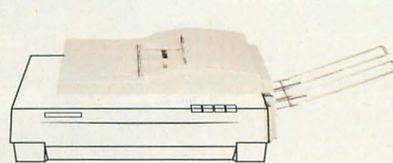
●シャープ独自の技術「デジタルズーム」搭載により繊細な線やズーム画像も忠実に再現。また「ワンウェイスキャン方式」を採用し、凹凸のある原稿も鮮明に読み取りできます。

通常の拡大時（当社従来機）　デジタルズーム（JX-330シリーズ）

透過原稿読み取りユニットとADFを同時装着できます。

透過原稿読み取りユニットは、35mm(ネガまたはポジ)フィルムからレントゲン写真まで各種透過原稿※4に対応。基本解像度600dpi/1200dpiの2種類をご用意しました。また最大50枚までの原稿を自動送りできるADFも同時装着できます。※5

X68000対応カラーイメージスキャナ

JX-330X

透過原稿読み取りユニット（オプション）
JX-3F6 標準価格 98,000円（税別）
JX-3F12 標準価格138,000円（税別）

カラーイメージスキャナ
JX-330X 標準価格178,000円（税別）

ADF［原稿自動送り装置］（オプション）
JX-AF3 標準価格 58,000円（税別）

使いやすい高機能画像入力ソフトを標準装備〈JX-330X〉
●Scanner Tool/S(画像入力ソフト)、対応フォーマット形式：ZIM、PIX、GL3、PIC、GLX、GLM

※1 2400dpiは当社独自手法による疑似解像度です。※2 読み取り開始から読み取り終了までの動作時間。ただし初期動作およびデータ転送時間を除く。(室温25℃)※3 クラスとは、A4フラットベットクラスのこと。'95年5月現在。
※4 読み取り可能なサイズは機種によって異なります。※5 ご使用になるアプリケーションにより対応が異なります。

■消費税及び配送・設置・付帯工事費・使用済み商品の引き取り費等は、標準価格には含まれておりません。

■資料のご請求・お問い合わせは　シャープ株式会社 プリントシステム事業本部プリントシステム営業部
〒545 大阪市阿倍野区長池町22番22号 TEL.(06)621-1221(大代表) FAX.(06)629-1207
〒162 東京都新宿区市谷八幡町8番地 TEL.(03)3267-4410(ダイヤルイン) FAX.(03)3260-2159

シャープ株式会社

SEGA SATURN MAGAZINE

セガサターンマガジン

NEXT GENERATION
SEGAGAME MAGAZINE
© セガ・エンタープライゼス

540YEN
好評発売中!!

[特集]
サターンでRPG! PART 2

6月号

アクレイム、カプコン、サターンタイトル発表!
ストリートファイターII ムービー
ストリートファイター リアルバトル オン フィルム
モータルコンバット 完全版
エイリアントリロジー 他

[AM2研EXPRESS NEO]
サターン版バーチャファイター2
パイの演武デモを徹底解析!! 1/60の動きに迫る!!

▼NEW RELEASE TITLE
最新のセガサターンソフトをキャッチUP!
新・忍伝／ツインビー対戦ぱずるだま／レイフォース

▼COMING SOON SOFT
発売目前！期待のセガサターンソフトを大紹介！
極上パロディウスだ!／スーパーリアル麻雀 PV
バトルファイター♥MIKU

▼SEGA SATURN COMPLETE GUIDE
発売後のセガサターンソフトを徹底攻略！
デイトナUSA／パンツァードラグーン／三國志 IV

▼HYPER MEGA EXPRESS
TEMPO／パラスコード／ライトクルセイダー

特報!
ワールド アドバンスド大戦略
バーチャコップ

お近くの書店でお求め下さい
ソフトバンク株式会社／出版事業部 販売局 TEL.03-5642-8100

お待たせしました!

あのスーパーリアル麻雀PVの公式設定資料集

ついに発売!

スーパーリアル麻雀 〜P.V〜 原画&設定資料集

▶田中良描き下ろしピンナップ付き◀

▶業界初(?)の飛び出す絵本も付いているのだ◀
(どんなモノかは見てのお楽しみ)

▶未公開設定資料原画とセル画が満載◀

▶みづき,綾,晶の3人のすべてがこれ1冊で全部わかる!◀

▶今回もバッチリ内容保証!◀

▶初回限定のおなじみプレゼントもあるぞ!◀

もはや麻雀ゲームの定番となったスーパーリアル麻雀シリーズ最新版PVの未公開設定資料満載。動画枚数1000枚突破のアニメーションシーンもバッチリ完全収録。おなじみのピンナップ付録に加え、巻末付録に"飛び出すPVポップアップ"が付いた今までにない充実度。買わないと一生後悔するかも!

MIZUKI

AYA

A4判
カラー80ページ
+
モノクロ32ページ
+
飛び出すPVポップアップ
+
とじ込みピンナップ
定価2,000円

AKIRA

好評発売中

シリーズ既刊◆好評発売中

スーパーリアル麻雀 PII&PIII ファンブック

A4判
定価2,000円

スーパーリアル麻雀 PIV 原画&設定資料集

A4判
定価2,000円

ソフトバンク株式会社/出版事業部 SOFT BANK
販売局 TEL:03-5642-8101

［第49回］
ピカピカ

江口　響子

響子inCGわ〜るど

JR中央線で，新宿から1時間ほど乗り八王子までくると，東京都といえど，のどかな風景が広がってくる。芽吹き出した淡い緑の木々から，白い山桜が謙虚に顔を覗かせている。対照的なのは桃の花。赤い色が目にまぶしい。

そんなのんびりとした里の春の風情を眺めていると，地下鉄サリン事件などで「最近の東京は治安が悪い」という評判が，うそのように思えてくる。

この4月から八王子にある東京造形大学という美術学校で，講師をすることになった。週に一度，金曜日の午前と午後，必修の2科目である。

初日，午前7時に家を出る。う〜ん，こんなに朝早く学校に行くなんて，高校の部活(陸上部だった)の朝練以来だ。電車を乗り継ぎ2時間弱で，学校に着く。

午前中は，1年生対象の授業。いままで，専門学校ではときどき教えていたが，大学での講義は初めてなので，こちらもまた1年生だ。

人数は40人弱。そんなに大人数でもないので，自己紹介を兼ねて，自分がいま，なにに一番興味をもっているか答えてもらった。

⋮

日本の古典芸能ならなんでも……なかでも能が好き(♀)
映画が好きで，タランティーノはもちろん好きだけど，100%好きというわけではない。「パルプフィクション」はもちろん見た(♂)
いまヒトラーの「我が闘争」を読んでいる(♀)
「JM」はウィリアム・ギブスンの原作とかけ離れていると思う(♂)
東西を問わず，若手のお笑いが好き……(♀)

⋮

趣味はゲームです……といった学生はいなかった。なぜかほっとした。また，X68000をもっている人もいなかった。これはちょっと残念。さらに学生の自己紹介は続く。

⋮

料理とケーキ作り，おいしいものを食べ歩くのが趣味(♂)……彼はなんだかOh!X編集長の前田さんに，優しそうな雰囲気がよく似ている。
盆栽を始めた……(♀)
1年間に100本ペースで映画を観てきた。ここ5年で500本ぐらいは観たかな(♂)
ミラン・クンデラを最近は読んでいる。「存在の耐えられない軽さ」とか「不滅」とか……(♂)

などなど。いったん社会人になってから入学した人，何年か浪人してからきた人，現役の人ももちろんいる。1年生といってもさまざまだ。

でも新しいことを始めようとしている人は，年

齢に関係なくピカピカに輝いている。これは皆同じ。ピカピカの１年生っていうキャッチコピーがあったけれど，小学生だけでなく，大きくなってもそのとおりなんだなあ。

担当の科目は情報処理演習というものだけど，コンピュータに触れるのは初めての学生が多いので，内容はごく簡単にするつもりだ。でも，一番大切なのは，このピカピカがなくならないようにすることかもしれない。

春は草や木がいっせいに芽吹き，本当の意味で１年の始まりを感じさせる季節だ。これから１年間，大学生活でどんなことがあるだろうか。

今回のCGデータ

1280×1024ピクセル
1670万色フルカラーを4×5ポジで出力
作成手順
XL/Imageのビームで背景の画像を作成。サイクロンでストライプの目の画像を作成。目の画像のαチャネルを用いて，目と背景をX68000上で合成。

SOFTWARE INFORMATION

ビデオゲームアンソロジーシリーズvol.13として「バラデューク」が移植決定だ！ 期待して待つことにしよう。そして，TAKERUから学園コメディ「EXCITINGみるく」の発売が正式に決定したぞ。

バラデューク

グロテスクながら美しい。そんな言葉がピッタリな迷路型シューティングゲーム，それがバラデュークである。10年前の1985年，それはカッコイイありがちなゲームに慣れていたプレイヤーの前に姿を現した。しかも斬新なゲームを送り込み続けてきたナムコからの衝撃的なデビューであった。

ルールは迷路化されたステージ内部を自由に移動し，青いモンスターを撃って全滅させていけばよい。ただ，ステージ構造が面により独特のものになっているので，ある程度の作戦というものが要求されるところが，このゲームを奥の深いものにしていた。しかし，最大の特徴は，やはりそのグラフィックと独特な雰囲気であろう。

いまでこそ，リアル調のモンスターやホラーの要素を意識したゲームは，掃いて捨てるほどある。しかし，当時はこのヌルヌルしたグラフィックの質感が非常に異質で，かつリアリティのあるものに映ったものである。こういったスタイルの元祖として，当時のプレイヤーの記憶に印象的に残り続けるバラデュークが，ついにアンソロジーシリーズとしてX68000の画面に再現されたのである。

とにかくドロドロした雰囲気が全面に踏襲され，敵の統一感も含めたイメージは，現在のゲームにも劣らないほどの緻密な完成度を持っている。敵モンスターに囚われている，黄色のかわいい仲間である「パケット」ですら，ほとんどコミカルな要素が感じられず，シリアスドロドロの世界の引き立て役になってしまっている。つまりゲームの中のどの要素を見ても，そこにバラデュークの世界そのものが凝縮されていると思っても間違いないであろう。

とはいえ10年も前のゲームなだけに，キャラクターの容量や種類，ゲーム自体の処理の複雑さといった部分を見ると，現在の豪華美麗大容量のゲームとは，大きな隔たりがある。しかし，ステージの迷路的構造，しかも敵の配置による事実上目に見えない壁や一方通行といった，プレイが進むたびにプレイヤー自身が発見したり，手痛いダメージで思いしらされるという，ゲームの面白さは，現在のゲームにも劣っていない。むしろキャラクターの少なさから，その構成の緻密さに感心するほどである。

最近のキレイでハデなだけのゲームにもの足りなさを感じている人には，もってこいのお勧めゲームといえるだろう。往年のファンは，文句なしに要チェック。いまからルーレットの練習をしておこう。 (八)

X68000用　5″2HD版　9,800円(税込)
電波新聞社　☎03(3445)6111

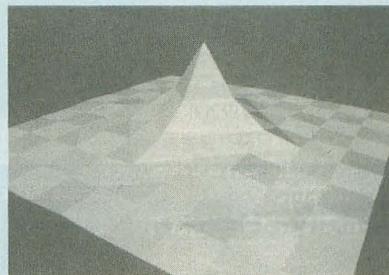
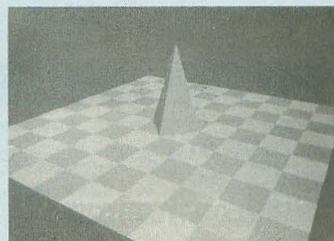
発売中のソフト

★学研統合電子辞書 for SX-Window国語・漢和辞書　TAKERU　4/18
X68000用　5″2HD版　9,800円(税込)
★学研統合電子辞書 for SX-Window英和・和英辞書　TAKERU　4/18
X68000用　5″2HD版　9,800円(税込)

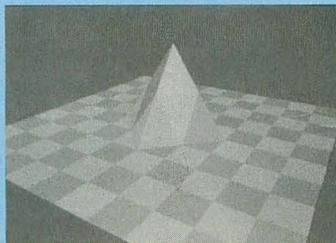
新作情報

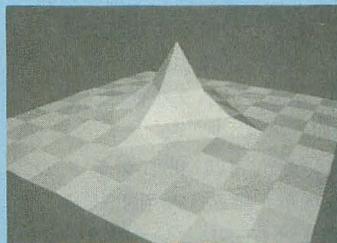
★バラデューク　電波新聞社　5/26
X68000用　5″2HD版　9,800円(税込)
★X CASE　Beシステム
X68000用　5″2HD版　19,800円(税込)
★Traum　象スタジオ
X68000用　5″2HD版　価格未定
★麻雀悟空・天竺への道　シャノアール
X68000用　5″2HD版　9,800円(税別)
★地球防衛MIRACLE FORCE　カスタム
X68000用　5″2HD版　価格未定
★プリンセスメーカー　ニュー
X68000用　5″2HD版　14,800円(税別)
★EXCITINGみるく　TAKERU　7/未
X68000用　3.5″/5″2HD版　1,500円(税込)

今回はCGAコンテスト入賞作品にも使われた物体変形ツールEXPOINT.Xを紹介します。なかなかやっかいな問題もありますが，がんばって特性をつかんでうまく利用していきましょう。

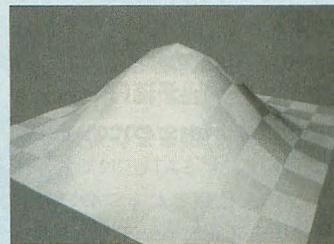
写真A　市松模様の画像

写真B
原点を(0, 0, 300)にもち上げる

写真D　原点を(300, 400, 300)にもち上げるアニメーション

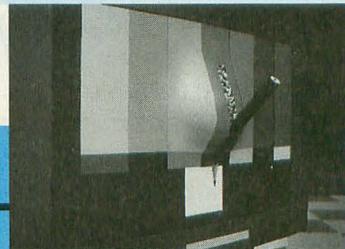
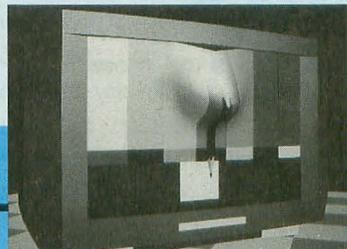
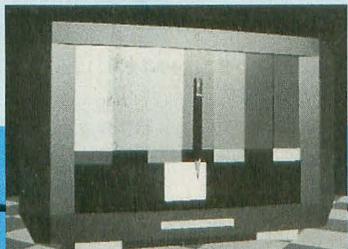
写真I　特定の点のみを引っ張る

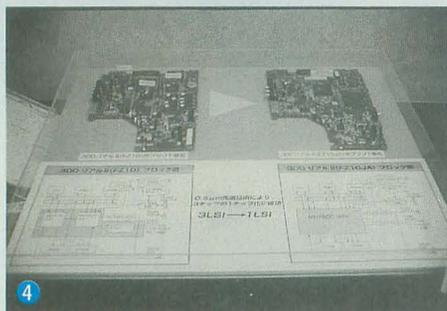
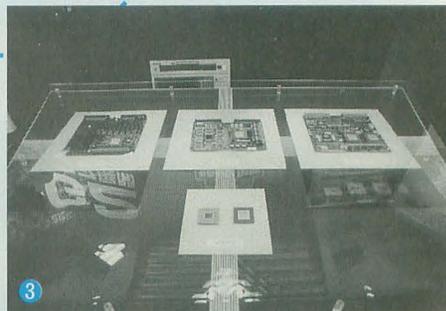
変形タイプ(膨らみ方の違い)

写真E1　比例(直線的)

写真F1　コサインカーブ

写真K
基本距離が小さくなるようにモデリングすると，盛り上がり方も滑らかになってくる

写真G1　放物曲面

写真H1　反比例

写真L
SHADE.Xで法線ベクトルを平均化。かなり滑らかだ

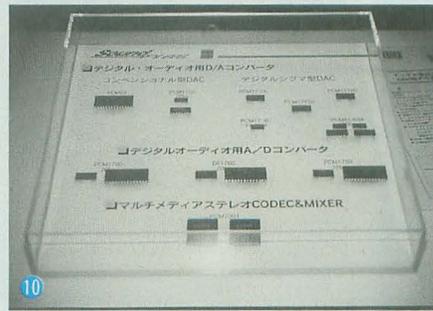
サンプルで作ってみました。テレビに映った鉛筆が画面から抜け出るところです

マイクロコンピュータショウ'95

micro computer SHOW '95

4月18日から21日までの4日間，東京流通センターでマイクロコンピュータショウ'95が開催された。スペースは年々縮小され，往時の半分以下の規模となり，今年の出展社は31社と少し寂しいものがある。

比較的元気のよかったのは日立製作所，松下電器産業など。SATURNでお馴染みのSH2チップの上位版となるSH3の発表を初め，SHシリーズをSuperHファミリーとして大々的に展示していた。SH3はキャッシュの大容量化，高クロック化などで60MHz時60MIPSを実現したプロセッサだ。それを使ったPDAの試作機も展示されていた。

松下電器産業はMPEG2エンコーダチップやリアルタイム色変換コントローラなどを展示。リアルタイム色変換コントローラはデバイスによる色表現の違いを簡単に吸収してくれる。ビデオ映像中の絨毯の色を変えたりなどといった応用デモはなかなか説得力がある。早くプリンタなどに内蔵してほしいものだ。

NECもV830など，100MIPSクラスのRISCチップを展示。組み込み用RISCチップは激戦だ。富士通もFR20というオリジナルRISCプロセッサを発表。2バイト固定命令はともかく，2.7V駆動と超低消費電力で，周辺をチップ内にまとめるなど，明らかにSH2を意識した構成になっている。用途として挙げられた家庭用ゲーム機の絵がSATURNそっくりなのがほほえましい。

その他ではフラッシュメモリなど，次世代デバイスの展示が目立っていた。

❶日立のSHシリーズ　❷MPEG2用エンコードチップのデモ　❸MPEG2チップの評価ボードだ　❹集積の進むREAL2の基板。右は1チップだ　❺色変換チップのデモ　❻V800シリーズも充実してきた　❼今年もTRONは健在だった　❽大容量マスクROM　❾ICカードによるヘッドホンステレオ　❿高性能になった各種PCMチップ

Open the SX-WINDOW

CONTENTS

SX-WINDOWシステム。
ネットワークに対応していない，仮想記憶を持たないなど，システム的な弱さを抱えてはいるものの，操作性ではもっと高い評価を受けてよいシステムである。
ウィンドウシステムの操作法に特に規定はない。別に，ファイルの起動で，マウスをちょっと後ろに引いたあと前に2回入れて両ボタン押し……といった操作でもかまわないのだが，人はやはり楽な方向のものを選ぶ。ダブルクリックすら最近は困難とされ，シェルからはなれたところでアプリケーション専用のランチャーを設けることが流行しつつある。おそらくダブルクリックの設定時間とマウスの移動速度（べらぼうに速くしたがる人が多い）が適切ならさほど困難なことではないのだろうが。
マルチタスクマルチウィンドウ環境は，まだまだ普通の人には使いきれないくらいのポテンシャルを秘めている。マニュアルべったりの使い方でもまだ奥が深い。それをさらに究めるいくこともできる。
カスタマイズによって自分の手に馴染む環境を作り出すということ，慣れるにしたがってマルチタスクマルチウィンドウという環境を生かした操作を体得するようになるということ。また，ちょっとしたアイデアでツールの活用範囲が広がること。単にアプリケーションを立ち上げる場としてのシステムではなく，そういった，いわば"OPERATING PLEASURE" の追求がいま始まろうとしている。
そういった局面ではSX-WINDOWのような素直な操作体系は最適である。
広がるのは前人未踏の領域かもしれない。タスク間に有機的関係を構築することにより，幾分なりとも眠っていたウィンドウ環境の可能性を引き出すことができるだろう。フォントなどの資源を充実させていくことで，各種ツールの活用の幅を広げることができるだろう。
こうしていちだんと快適な環境が築かれる。さらに使いこなしていくことでユーザー自体がウィンドウ環境へ適応していく。そのためには現システムを正しく評価しなおすことが重要であろう。

活用の基礎知識
ウィンドウ環境への招待

Nakano Shuichi 中野 修一

なぜ，わざわざ重いSX-WINDOWを使うのか?
その答えはマルチタスクマルチウィンドウ環境にある
ここでは人にやさしいウィンドウ環境のあり方を考えてみよう

SX-WINDOW環境を選択することは，X68000の有するグラフィック機能のほとんどとスプライト機能などを捨て去ることでもあります。それは，純粋にメモリ量とCPU速度で勝負する世界——そういう分野で見ればX68030とて3世代前のハード仕様にすぎません。

しかし，幾多の不利な状況にもかかわらず，SX-WINDOWはWINDOWSや漢字TALKに比べてもかなり使いやすいシステムとして存在しています。

考えてみればSX-WINDOWというのはごく当たり前の環境を揃えているだけです。世の中，パソコンがGUIで使いやすくなったとかいっていても，ごく基本的なことすらやっていないということになります。

巷では，Macintoshを「使いやすいから」という理由で選ぶ人がいるくらいですから，少々重かろうがSX-WINDOWを使う理由は十分といえるでしょう。

環境さえ整えれば，SX-WINDOWは実に快適なプラットフォームになります。コンソールが装備されたことで，私は最近ではほとんどSX-WINDOWから抜けることもなくなりました。SX-WINDOWから外れるのはグラフィック関係の処理をするときくらいですね（SX-BASICでやってたら画像処理に半日かかった……）。

まあ，私個人がSX-WINDOWを使っているのは，X68000史上で最高のワープロであるシャーペンを使いたいからというのが最大の理由ですが，このシャーペンのためだけでも従来の環境を捨てる価値はあると思います。ここではSX-WINDOWを使ううえで，知っておいたほうがよいちょっとしたことをまとめてみましょう。

SXのユーザーインタフェイス

SX-WINDOWのディレクトリウィンドウ周りは実に素直にできています。ファイル操作に関しては，足りないものはおそらく「ファイルの生成」くらいしかないでしょう（いきなり新規ファイルを生成することができない）。

●基本はCOPYALL

SX-WINDOWでときどきディレクトリのコピーに，新しいディレクトリを作成して，ウィンドウを開き中身を全選択して持っていく……といった操作をしている人がいますが，アイコンの移動は基本的にCOPYALLですから，ディレクトリごとアイコンを運べばすべて解決します。バックアップMOに作業ディレクトリをコピーするのも，ディレクトリごとドライブアイコンに放り込めば変更のあったファイルだけファイルを自動的に更新してくれます。基本ですね。

●ドラッグ＆ドロップ

アプリのウィンドウにアイコンを放り込むと，そのファイルを読み込みます。前のデータを廃棄してしまうのがイマイチですが(新規オープンの選択もほしかった)，これは非常にわかりやすいインタフェイスです。

またダイアログなどの文字入力部分にアイコンを放り込むとファイル名が渡されます。コンソールでデータファイルを指定する際などは便利です。

ただ，昔からいっているように，ドラッグという作業は距離が長くなると誤操作が多くなる処理です。ドラッグなしで同等の動作を実現する方法も考えるべきでしょう。個人的には左プレスで「つかむ」処理を行うのが妥当かなと思っています。

ちなみにMacintoshでドラッグ＆ドロップといえば，実行ファイルのアイコンにデータのアイコンを重ねると，データファイルを読み込んでアプリケーションが起動する，というものです（最近はほかの意味もありますが)。

これが，なぜ便利がられているのかよくわからなかったのですが，Macintoshではアプリケーションのウィンドウにアイコンを放り込めないと聞いて，ようやく納得できました。

Macintoshを初め，ドラッグ＆ドロップは，これからあちこちで流行しそうなインタフェイスとなっています。早くドラッグの「使いにくさ（わかりにくさではない)」をちゃんと理解する人が現れてほしいものです。

●ポップアップメニュー

ファイルに対して，それを読みたいと思ったときや，サイズを確認したいと思ったとき，ダンプしてみたいと思ったときで，好きな動作を指定することができます。ツールを起動せず，操作したいデータの位置で処理できるというのが特徴です（当たり前のことですが……)。

シフトキーでメニューの内容が変わることもあります。シフト時の動作が規定されているかどうかが見てわからないのがナニですが，試さないと損をすることもありますのでメニューの出るところでは片っ端からシフト併用を試してみるのもよいでしょう。

なお，2ボタンマウスを使っておきながらいまさらという感じはしますが，WINDOWS95では右クリックでポップアップメニューが出るようになったそうで，これはMacintoshに対する完全な優位点だそうです。

●エラーダイアログ

知らない人が多いようですが，エラーダイアログの「実行」「取消」ボタンは，マウスをわざわざ運ばなくてもカーソルキーで選択することができます。「確認」などは初めからボタンの周りがへこんでいることがあります。これならリターンキーを押すだけで済みます。どうせなら，すべてのダイアログで実現していてくれればもっとよかったのですが。

一方，WINDOWSやMacintoshでは「マウスカーソルが走っていく」とか，「ボタンのところにカーソルがジャンプする」といったものが重宝されています。

SX-WINDOWでは基本理念として，「マウスカーソルはユーザーの操作以外では移動しない」というのがありますから，これは採用できないインタフェイスです。キーボードでの操作もよいのですが，私の経験でいえば，ダイアログではMATIER式のボタン割り当てがもっとも使いやすいと思われます。大切なのはカーソルを操作することではなく，システムに意思を伝えることですから。

●日本語処理

必ずASK3を使い，なるべくSXKEY.Xを登録してください。

変換作業時にESCで初期状態へ戻すのは当たり前ですが，シフト＋ESCで，直前に日本語FPに渡した文字列を呼び出すことができます。UNDOキーでは直前に確定した文字列を変換し直すことができます。ちょっと長い文字列を打ち込んでいて，誤って確定してしまった場合には便利です。

例を挙げてみましょう。

　　すもももももももものうち

と打ち込んでいて，誤操作で変換範囲を間違えて，

　　李桃も桃も農地

のまま確定してしまったとしましょう。この場合，UNDOで，

　　李桃も桃も農地

のまま変換状態に戻せますし，さらにESCで，

　　すもももももももものうち

という最初の入力文字列にまで戻せます。ここでESCを押しすぎると文字列が全部消えてしまいますが，その場合はシフト＋ESCで呼び出すわけですね。

まあ使っている人にとっては当たり前の機能ですが，シフト＋ESCなどは文字列コピー代わりに使ったりもできて便利です。

シャーペンのインライン変換とSXKEY.XによってSX-WINDOWの日本語処理もずいぶん高度になりました。システムレベルでインライン変換できないのは問題ですが，シャーペンのデキがいいので，とりあえず大きな障害にはなっていません。

あえて改善すべき点を挙げれば，ウィンドウごとに日本語フラグを持って，フォーカス時に日本語FPのON/OFFを自動でやってほしいということくらいでしょう。あっと，無論，ASK3自体の改善も必須ですけど。

マルチウィンドウ環境の危機

最近Macintoshのバージョンアップとか，WINDOWSのフリーソフトで「便利！」といわれているもののなかに，デスクトップ上に付箋のようにメモを張り付けるというツールがあります。

最初に思ったのは「なぜデスクトップに余白が空いてるんだ？」ということでした。

私の使っている環境では常にデスクトップの８割以上はウィンドウで覆われています（わざわざ背景グラフィックを見るモードをつけているくらいです）。もちろんSX-WINDOWですから，使えるツールというのは決して多くありません。しかし，デスクトップには常にウィンドウが大小あわせて15個くらい，タスクとしては40～50タスク走らせて使用するのが当たり前になっています。ディレクトリウィンドウがいくつかと，シャーペンが３つくらい，コンソールにSX-BASICや時計などの小物……とこれくらいの環境です。

画面を見た国民機ユーザーの人からは「そんなにいじめなくても……」といわれたこともありますが，私としては特に重いことをやっているつもりはなく，SX-WINDOWも軽快に動作しています。

聞いてみると，WINDOWSでもMacintoshでもエディタとDOS窓くらいしか使っていないとか，複数のアプリを立ち上げること自体が少ないということのようです。

メモを取りたければ，常にエディタを開いておけばすむはずですが（私はそうしてる），わざわざデスクトップに張り付ける，すなわち，それが見える状態にある，すなわち，ウィンドウはあまり開かない，ということを示しています。うーん，いったいなんのためにウィンドウ環境を使っているのでしょうか？

さらに気になるのは，次世代のマンマシンインタフェイスとして各所で開発されているものです。

雑誌などで見たことがある人もいるでしょう。画面にオフィスの絵があって，そこに書かれた絵を使ってオペレートしようというようなものです。メモをクリックすればメモ帳，電話をクリックすれば通信，といった具合にアプリを起動します。MacintoshもWINDOWSもその他のOSも軒並みこういった方向性に走っているようです。

ここでも疑問になるのは，なにかアプリを起動したら，ほかのアプリはどうやって起動するのだろうかということです。画面切り換えでアプリランチャーの画面が独立しているというのならよいのですが（それはそれでうざったいが），でないとマルチウィンドウ環境というのは生きてきません。

おそらく基になったのはビル・アトキンソンの作ったMagicCap（PDA用のOS）だろうと思いますが（あまり詳しく調べていないので断言はできないが），これはシングルウィンドウ環境のために作られたものだと思います。なにも考えずにマルチウィンドウ環境へ応用しようというのには無理があると考えるべきでしょう。

不満を持つということ

SX-WINDOW関係の記事では，私はたいてい文句をつけていますが，前述のとおり私の周りはもうSX-WINDOWしか使わない環境になりつつあります。

BTRONにも触ってみましたし，WINDOWSが予想以上に理解不能であることや，NeXT STEPが意外にわかりにくいこと，やはりMacintoshはわかりやすいが使いにくいことなども確認してきました。

SX-WINDOWはシステムの美しさと使いやすさを両立させた数少ない例といえます。もともとアップル系のインタフェイスを導入しておきながら，最近はマイクロソフト系のインタフェイスを追加してくるなどというわけのわからない部分も多少ありますが，その基本部分はいまだに他のシステムを寄せつけないものがあります。純国産システムとしてもがんばってほしい気持ちはあります。

しかし，システム面での弱さはいかんともしがたいものがあるのも事実です。ネットワークの対応が皆無なことや，印刷システムの未熟さ，開発環境の整備遅れ，システム情報の不透明さ，アプリの少なさなど，挙げればきりがないでしょう。

ものごとを進化させていく過程は往々にして，なにかに不満を持ったりすることから始まります。そういった意味で不満を持つことは大いに重要なことだといえるでしょう。世間を見れば，どうしようもなかったWINDOWSでさえ徐々に改善されつつあります。進化の速度からいえばいつまでもタコ呼ばわりはできないのかもしれません。現状に甘んじていることはできないのです。

与えられた環境に対して，どうも嫌だとか，使いにくいと思ったら，それがなにに由来しているのか，どうすれば解決できるのかを考えていくことが重要でしょう。

導入のための初歩の初歩

SX-WINDOWを始めてみよう

Taki Yasushi 瀧 康史

「SX-WINDOWなんて」と食わず嫌いをしてる人も多い
なんといっても，まずは触ってみることが肝心
そこでインストールから始めるSX-WINDOWの使い方を紹介する

SX-WINDOWの世界へようこそ！

今回は超初心者向けのお話です。SX-WINDOWをインストールし，操作を覚えていくことから始めましょう。

＊　＊　＊

私はいま，原稿を書く仕事をしています。原稿を書くために，デスクトップ（SX-WINDOWの画面の上）にはシャーペン（SX-WINDOW標準添付のエディタ＆ワープロ）があります。こうして原稿をタイプしている最中，図がほしくなってしまいました。そのためにドローイングソフト，Easydraw（別売のドローイングツール）を起動します。メモリが不足していない限り，シャーペンを閉じることはなく，新規にウィンドウ(Easydraw)を開くことができます。

これが，SX-WINDOWを導入する前の場合なら，WP.X(XVI以前に添付されていたワープロソフト)で原稿をタイプしていて，ふと図を描きたくなったなら，原稿を印刷して終了し，CANVAS PRO-68K（Human68k上で動くドローイングソフト）でも起動しなくてはならなかったわけです。印刷するのはなぜかというと，それはもちろん，「本文と照らしあわせて」図を描くからです。

私のライターという仕事はあまり一般的じゃないかもしれません。しかし，普通の人も最近はワープロを結構頻繁に使うでしょう？　ワープロ文書の中に図を入れたいときはたいてい本文を説明する図を描くわけですから，どちらも見ながら，どちらも改変できるような環境でいるのがベストなわけです。SX-WINDOWならそれができます。文章を書きながら，図を張り込む。そういった操作ができるのがウィンドウシステムというものなのです。

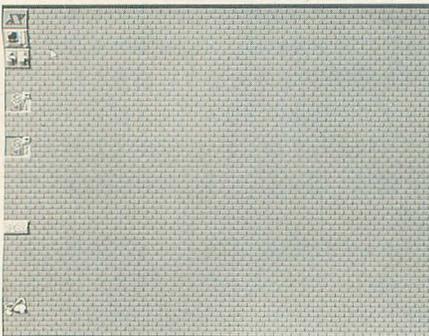



写真1　起動直後の画面

それでは実際にSX-WINDOWを使用してみましょう。まずはSX-WINDOWのシステムディスクの中から，システムディスクと，拡張ディスクを取り出します。システムディスクをドライブ0に入れ，拡張ディスクをドライブ1に入れて，FDからシステムを起動してください。

起動したデスクトップは写真1のようになっているはずです。こみいったことをするためには，インストール（以下の項で説明）という作業をしなくてはなりませんが，この状態でもある程度SX-WINDOWを楽しむことはできます。

正しく起動できたなら，マウスを動かすことにより，矢印のようなものが動くはずです。これをマウスポインタと呼びます。マウスポインタは名前のとおり，マウスの動きをそのまま画面に伝えるものです。システムにより，マウスカーソルとも呼ぶため，人によってはSX-WINDOWの世界でも，マウスカーソルと呼ぶことがあります。

マウスポインタは通常，自由に動かして構いません。マウスポインタを動かすだけでは，なにも起こらないからです。マウスポインタを通じてSX-WINDOWに指令を出したいときには，ボタンを押します。右ボタンと左ボタンがあるはずです。

なにもないところでマウスの右ボタンを押しっぱなしにします。この操作を右プレスと呼びます。するとメニューが現れるはずです。これをポップアップメニューと呼びます。右ボタンを離すとメニューは消えます。右ボタンを押したままマウスポインタを移動し，項目の上まで持っていくと，ポインタの示している項目の色が変わるはずです。これが選択している状態で，この選択状態で右ボタンを離すと，その命令が実行されることになります。

プレス以外にもSX-WINDOWではいくつかの操作体系があります。以下はマニュアルの8ページにあるものの引用です。

1)　ポイントする
　ポインタを絵の上に重ねる。
2)　クリックする
　ボタンを押して離す。
3)　ダブルクリックする
　ボタンを続けて2回押して離す。
4)　プレスする
　ボタンを押し続ける。
5)　ドラッグする
　ボタンを押したまま移動する。

以上，5種類の操作を組み合わせてSX-WINDOWに命令を与えるわけです。詳しい説明はマニュアルの9～10ページにあるのでぜひ読んでみてください。

さて，画面左にはアイコンが並んでいるはずです。左上から順に，Xマークのシステムアイコン，X68000の絵が描かれたデスクアクセサリアイコン，そしてページアイコンです。この下に，ドライブアイコンが2つ，A，Bと書かれて置いてあるはずです。FDD（フロッピーディスクドライブ）ですから当然，FDの形をしています。この下に掃除機の形をしたクリーナアイコンと，X68マークがあります。

左クリックで選択できるアイコンは，ドラッグして場所を移動することができます。こうして，デスクトップ中の好きなところに好きなものを置くことができるのです。選択できないアイコンでも，OPT.1キーを押しながらなら，選択することができ，ドラッグすることができます。

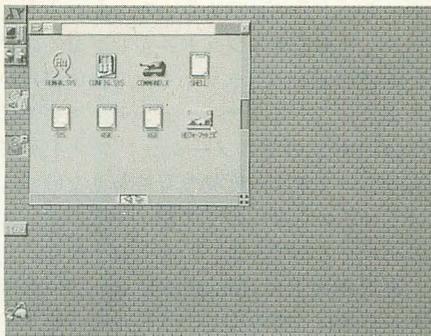



写真2　Aドライブの内容を見る

システムアイコンは通常緑色ですが，残りメモリが少なくなり始めると，黄色，赤へと色が変わっていきます。

ページアイコンはちょっと普通とは違うアイコンで，アイコンの右側，左側を左クリックすることにより，ウィンドウの優先順位を変えることができます。小さなウィンドウが，大きなウィンドウの下に隠れてしまったときに便利です。右プレスすることにより，開いているウィンドウのメニューの一覧が出てきます。もちろんここから選択もできるわけです。

ではここで，Aドライブアイコンをダブルクリックしてみましょう。写真2のようなウィンドウが開くはずです。これをディレクトリウィンドウと呼びます。SX-WINDOWでの主なファイル操作は，これを利用することになります。逆にSX-WINDOWシステムのファイル操作を覚えるにはこのディレクトリウィンドウの利用方法を覚えればよいことになるわけですね。

インストールから始めよう

「インストール」とは，ぶっちゃけた話，「使用できる環境にする」こと。というのも，市販されているソフトウェアの多くが，最近は「フロッピーディスクを入れてすぐ起動」ではなくなってきているからです。この理由として，まず挙げられるのは，システムそのものが非常に大きくなっていて，FD1枚～2枚の中には入らなくなってきていることです。ただこれは建て前のようなもので，実際の理由はほかのことにあります。それは統一された実行環境を作ること，これが大前提です。

まずはあなたが自由にカスタマイズして使える環境を作りましょう。この作業は，あなたがいままで，どのような環境でパソコンを利用していたかで微妙に異なってきます。

以下，SX-WINDOW ver.3.1を持っていることを前提にします。SX-WINDOWはver.3.0の段階でかなり成熟していますが，新しいバージョンに換える場合，インストールからやり直すことになるので，できればいちばん新しいSX-WINDOWver.3.1を購入してくるべきでしょう。ver.3.0を持っているユーザーならば，ユーザー登録ハガキさえ出していれば，バージョンアップの案内がきているはずです。9,300円でバージョンアップできるはずなので，ぜひ，バージョンアップの申し込みをしましょう。

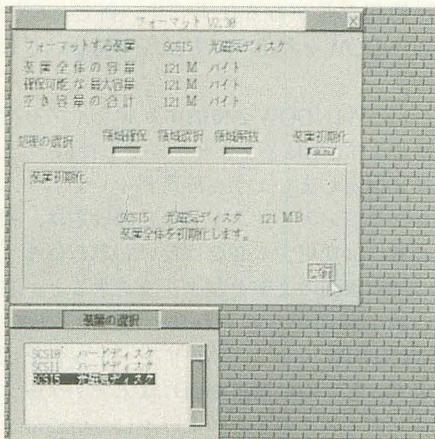



写真3　HDのフォーマット

それではインストールです。

まずは，新しくX68030＋HDD(ハードディスクドライブ)を購入した方，または新規にSX-WINDOWをHDDにインストールされる方から始めましょう。

●HDDのフォーマットから

HDDのフォーマットは簡単です。先ほど，Bドライブアイコンのディレクトリウィンドウを開きましたが，これを閉じ，Aドライブアイコンを開きます(写真2)。ここにHDフォーマット.Xがあります。ファイル名の最後が，「.X」のものは，ダブルクリックすると実行できるファイルです。これを起動します。さすがに写真撮影のために，編集部のHDDをフォーマットするわけにはいきませんから，代わりにMOをフォーマットしてみましょう。

まずこのウィンドウで行うことはHDD，またはMOの初期化です。検索されたIDの装置のなかでフォーマットしてよいドライブを指定し，写真3のような画面を出します。

HDD/MOはまず最初にフォーマットが必要です。フォーマット済みの場合，HDフォーマット.Xでは再フォーマットできません。

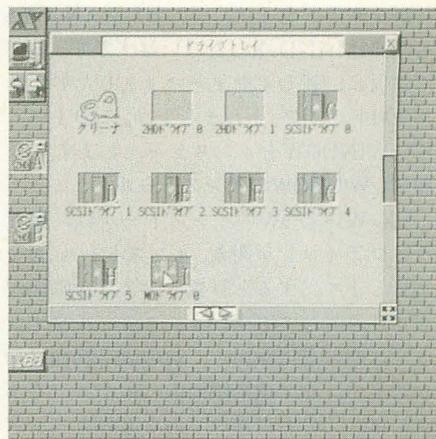



写真4　ドライブトレイを開く

領域確保は，領域確保のボタンを左クリックで設定して行います。すべてを一括で利用したい場合，そのまま領域確保を実行すればOKです。このとき，必ずシステムを転送するようにしてください。

初期化が終わり領域確保したら，HDフォーマット.Xを終了してください。終了はウィンドウの右肩にある×マークを左クリックします。自動的にリセットするはずです。

リセット後，再びフロッピーディスクから起動します。起動したときに，システムアイコンを右プレスし，「ドライブトレイ」をセレクトしてください。すると，写真4のようにFDD以外のドライブのアイコンが表示されます。編集室の環境の場合，このようにたくさんのドライブがあります。これらは個人個人のハードウェア環境によって違うはずです。

ドライブトレイにあるアイコンは，外にあるドライブアイコンと同じように利用することができます。ドラッグすれば外に出すことができます。ここではダブルクリックでオープンし，先ほどフォーマットしたドライブを探しましょう。フォーマットし

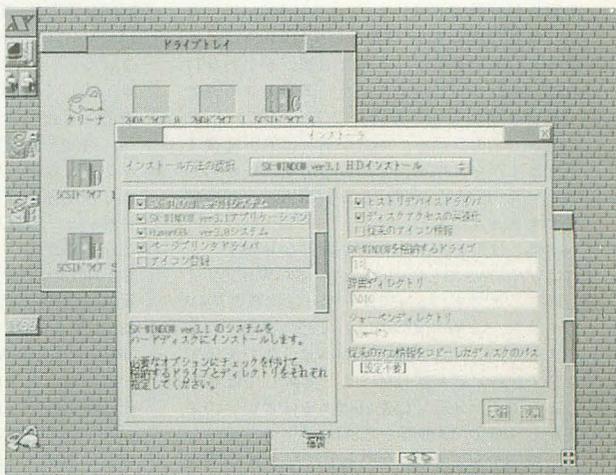



写真5　インストーラを立ち上げる

たばかりのドライブには，HUMAN.SYSとCOMMAND.Xしかないはずです。ここでは仮に，新しくフォーマットしたドライブを「Iドライブ」としましょう。このIに，SX-WINDOWをインストールします。

●SX-WINDOWのインストール

SX-WINDOWのインストールは，Bドライブアイコンを開き，インストール.Xを実行します。すると写真5のような画面が表示されるはずです。インストールウィンドウは左の選択画面と，右の設定画面に分かれます。左の選択画面の□のチェックボタンは，□の右に書いてあるものをインストールするかどうかを示すチェックボタンです。マークをつけると選択されます。

最初はSX-WINDOW ver.3.1システムが反転しているはずです。この状態では右の設定画面に，ヒストリデバイスドライバ，ディスクアクセスの高速化，従来のアイコン情報などがあるはずです。新しくSX-WINDOWを利用する方は，関係ないでしょう。

SX-WINDOWを格納するドライブを指定します。先ほどフォーマットしたIドライブを指定しましょう。辞書ディレクトリ，シャーペンディレクトリなどの設定は，初心者には変更する必要はないと思われます。

選択画面の文字を左クリックすると，右の設定画面がそれぞれ変わります。SX-WINDOW ver.3.1アプリケーションを設定すると，右の設定画面にはアプリケーションディスクのインストールについてのチェックボタンがあるはずです。アプリケーションディスク1にはSX-WINDOW必須のアクセサリが入っています。インストールは必ず行いましょう。

アプリケーションディスク2にはフォントとグラフィックのサンプルが入っています。人によってはいらないかもしれませんが，とりあえずインストールしておくのがよいでしょう。グラフィックの扱いがわかりますし。アプリケーションディスク3にはCGA形式のアニメーションファイルが入っています。一度見たら終わりなので，ほとんどいらないのですが，一応，インストールしてみてもよいかもしれません。

選択画面上のHuman68k ver.3.0システムを選択してみましょう。おそらくフォーマットしたばかりでしょうから，これもチェックしておきます。右の設定画面では，Human68kを格納するドライブの指定があります。この場合，J:あたりがデフォルトで指定されていますが，今回Iドライブでフォーマットしてしまったと思われるので，I:にでも直してください。

ページプリンタはPostScriptプリンタ，エプソンのESC/Pageプリンタ，キヤノンのLIPS3プリンタに対応しています。要するにレーザープリンタの類でして，つまり以上に挙げた3系統のレーザープリンタ以外はSX-WINDOWでは使えないということになります。

HDDにインストールする場合，アイコンの登録はデフォルトなので，する必要がありません。

この状態で右下の実行ボタンを押せば，インストールは始まります。あとは，手順に従って，FDDを抜き差しし，インストールを行ってください。

HDDインストールが終わったら，インストールしたHDDから起動してみましょう。正しくインストールできていたら，SX-WINDOW ver.3.1が起動するはずです。

ファイル操作の達人になろう

この調子で進んでいくと，いつまでたっても説明が終わらないので，このあとの操作は急ピッチで進めていきましょう。すでに「あなたの」SX-WINDOWがインストールされたので，好きに操作して構いません。自分なりに触ってみて，SX-WINDOWの利用方法を探索してみてください。

さて，SX-WINDOWではファイル操作を簡単に扱うことができます。ファイルの削除，コピー，移動，さまざまです。これらの基本は，ファイルアイコンをドラッグすることによって行います。

ポイントはアイコンをドラッグするということです。

たとえば，ファイルa.txtをAドライブからBドライブにコピーしたい場合，a.txtを

SX-WINDOWの導入に必要なもの

必要なものはもちろんX680x0シリーズ。10MHzシステムでは，残念ながらSX-WINDOWは重いのが実情です。今回，別のページで紹介したXellent30sをつけると，多少軽くはなりますが，こみいったことをすると，すぐに重さが気になってしまいます。

XVIレベルになると，VRAMが軽くなったこともあってか，こみいった処理をしなければ，重さはあまり感じません。Xellent30を搭載するとさらに多少速くなります。RED ZONE(*1)のレベルまで行くと，ほとんどの処理は重くなくなります。

そしてX68030，X68030 Dash(*2)になると，まず重さを感じさせなくなります。ただ，SX-WINDOWは通常の利用時の重さと，ひときわ重い処理をしたときの重さが，かなり違います。重くなる処理は，ちょっとばかり速くなったくらいでは解決できるようなの重さではありません。たとえばEasydrawのオブジェクト数が多量に増えたとき，XDTP，シャーペンによるベクトルフォントの利用などの処理はX68030をもってしても重くなります。

ところが040turbo(*3)を利用すると，これらの重さが一気に激減します。まさに040パワーといったところですか。理想をいえば040がほしくなってしまいますが，それでも贅沢をいわなければ，XVIレベルならばだいたい満足できる速度で動きます。なお，最近は7万円台でX68000 CompactXVIの新品を，中古があれば6万前後で買えますから，手を出してみるのもよいかもしれません。ただし，CompactXVIにはXellent30は接続できません。

本体とは別に，メモリも必要になります。これはWINDOWSでもMacintoshでもSX-WINDOWでも同じことです。X68030ユーザーは標準実装4Mバイトで，SX-WINDOWのほとんどの機能が利用できます。しかし，内蔵で12Mバイトまで実装できますから，迷わず12Mバイト実装してほしいところです。メモリはあればあるほどよいのですから。I/O DATA機器から8Mバイトメモリモジュール(SH-5BE4-8M)が売られています。実売5万円前後なので購入に踏み切るのもよいでしょう。

XVI，CompactXVIユーザーの場合，本体標準実装メモリは2Mバイトで，内蔵8Mバイトまで実装できます。TSRよりXSIMM VIとXSIMM VIcが発売されています。メモリなしで1万3千円ぐらいで売られていて，これに60nsの8MバイトSIMMをつけるのが理想です。本体の都合上，SIMMは8Mバイトでも6Mバイトまでしか認識できません。あわせて8Mバイトのメモリがあれば SX-WINDOWはほぼ満足に利用できます。

それ以外の機種の場合は，多少XSIMM10などを使っていただければ手軽にメモリ実装できます。

SX-WINDOWを利用するうえで，私からの推奨環境は，本体はXVI以上，メモリは8Mバイト以上といったところでしょうか。それ以下の装備でSX-WINDOWをヘビーに使っている人も存在しますが，多少，処理のうえでストレスがたまってしまうかもしれません。なお，ハードディスクはほぼ必須と考えていいでしょう。

＊1 満開製作所による，X68000 Compact XVIの24MHz改造品，またはその同等品を指す。ノーマルのXVIに比べると，あらゆる面で高速さを感じさせる。

＊2 同じく満開製作所によるX68030 33MHz改造品，またはその同等品を指す。X68030に比べ，速度を計測するベンチマークテストでは確かに速くなってはいるが，通常の使用ではあんまり速度差を感じない。ただし33MHzマシンから25MHzに戻ると，それなりの重さを感じることがあるが。

＊3 BEEP's氏における68040アクセラレータ。現在，計測技研より発売されている。CPUが68030から68040へ変わった違いは，インテルCPUの80386から80486DX2ぐらいの違いがある。10万円弱とお値段は高めだが，これによる高速化はだいたい3倍前後。X68030-33MHzマシン＋040turboの速度はすでにノーマルの68030の速度とは別次元を思わせるものがある。

ドラッグして，Bドライブのドライブアイコンまで持っていきます。これが基本操作です。Bドライブではなく，Bドライブの中にあるディレクトリ内に持っていく場合も同じです。ファイルをドラッグされたアイコンは，ディレクトリならば，フォルダが開いた形に変形します。

とあるディレクトリから，同一ドライブのディレクトリにドラッグした場合，コピーではなく，移動することになります。移動というのはつまり，コピーのあと，元のファイルが元の位置から消えてしまうということですね。同一ドライブ間で移動ではなくコピーしたい場合，CTRLキーを押しながらドラッグします。これは少し特殊な機能です。

SX-WINDOWではディレクトリとファイルは特に区別されていません。これが，SX-WINDOWのファイル管理システムの便利な部分です。ディレクトリ名は，ディレクトリ以下のサブディレクトリもすべて含みますから，ディレクトリそのものをドラッグして，いずれかのドライブアイコン，ディレクトリフォルダに持っていった場合，ディレクトリ以下すべてをコピーないしは移動することになるわけです。この作業は複数のディレクトリウィンドウにまたがっても構いません。

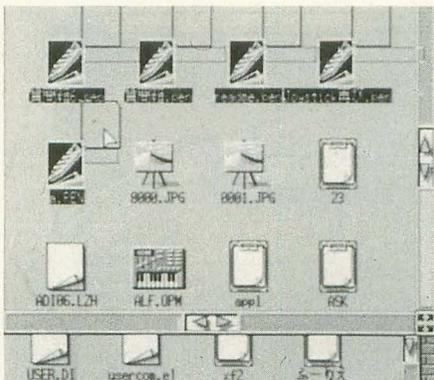
ファイルの削除は，クリーナアイコンを利用します。これは特殊なアイコンで，ファイルをドラッグしていくと，ファイルが削除されるものです。

ファイルの基本操作はこれだけです。

もしも，複数のファイルを同時にドラッグしたい場合，複数のファイルを選択すれば可能です。たとえばワイルドカードで＊.penをすべてドラッグしたいなら，ディレクトリウィンドウ中のファイルアイコンのない場所で右プレスによるポップアップメニューを出し，ワイルドカードを選択して，打ち込みます。こうして＊.penと入力すると，＊.penのファイルすべてが選択され，反転することになります。選択されたひとつのファイルをドラッグすると，それらはすべて道連れのようにドラッグされていきます（写真6）。

複数選択する場合はこれ以外に，シフトを押しながらファイルアイコンを左クリックすると，前に選択されたファイルをキャンセルせずに，選択ファイルをどんどん増やしていくことができます。また，ポップアップメニューの「すべてを選択」を選んだり，写真7のようにディレクトリウィンドウ中のアイコンのないところで左プレス

写真6　複数ファイルの移動

写真7　矩形領域の範囲指定

し，ポインタをそのまま移動したときにできる矩形で指定などさまざまです。

複数選択されても，それがディレクトリでも，ファイルの移動，コピー，削除が同じようにできるのがディレクトリウィンドウの特徴であり，このディレクトリウィンドウが複数開き，どこにでも持っていけるというのが，SX-WINDOWの直感による操作のしやすさといえるでしょう。

ファイルのドラッグはファイルのメンテナンス以外でも行います。たとえばシャーペンのウィンドウ内に，なんらかのファイルをドラッグしていくと，編集ファイルを破棄して読み込もうとしますし，シャーペンが新規編集のダイアログを出したときに，複数のファイルを持っていくと，ファイル名を直接，複数個与えることができます（写真8）。

このほかにも，キャンバス.XにPICファイルを持っていくと，画像ファイルをロードすることができますし，SX-PhotoGraphyに至っては，CD-ROMドライブアイコンまでドラッグします。これとはまた別に，ドライブアイコンを別のドライブアイコンに重ねると，コピーオール（ファイルの中身をすべてコピーする）や同じタイプのディスクなら，ディスクコピーなどができるようになっています。

ファイルアイコンのドラッグにより，アプリケーションがファイルを読み込むのは，SX-WINDOW操作の基本中の基本です。ときどきそうプログラムされていないアプリケーションもありますが，とりあえず，対応しているファイルをドラッグして放り込んでみるのは，SX-WINDOWの試行錯誤では必須のことです。

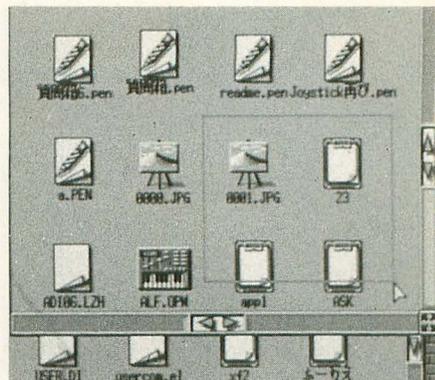



写真8　これらのファイルを一度に開く

メニューをオリジナルに！

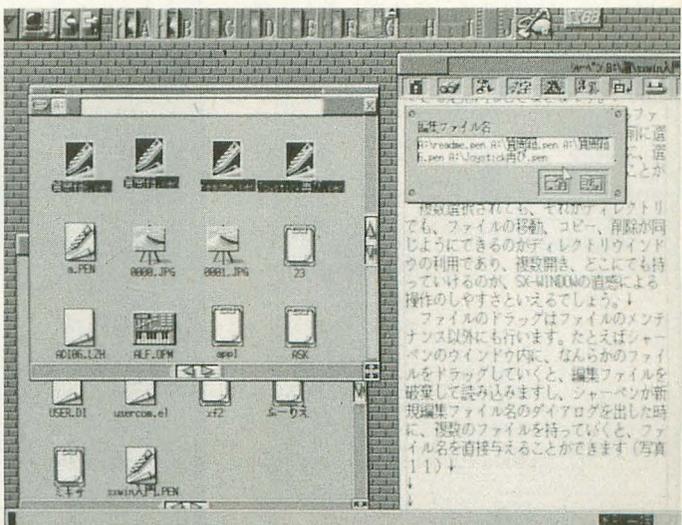
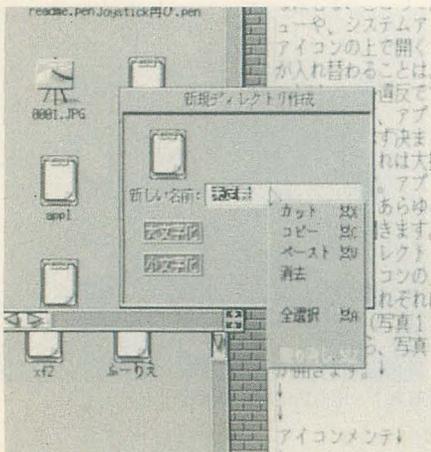
SX-WINDOWにとってメニューは大切な役割を持っています。SX-WINDOWではすべてのアプリケーションが同時に動いているわけで，すべてのアプリケーションが「自分のウィンドウ内」という決まったテリトリー（領域）を持っています。

その場所でやっている作業に対する処理がメニューで選択できます。

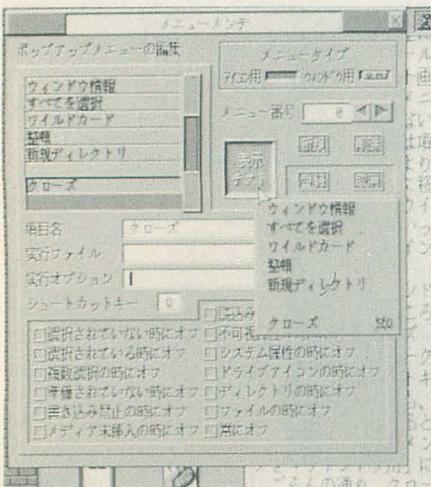
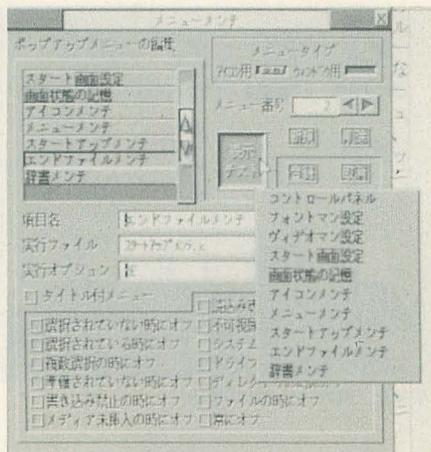
文字入力があるところでは入力文字列，ディレクトリウィンドウ内ではそのディレクトリに関する操作，ファイルアイコンではそのファイルに対する操作がメニューで出てきます。

同時に，メニューは同じ場所で，単に右プレスしたとき，キーボードのシフトキーを押しながら右プレスしたときの2種類があります。ソフトの設定によっては，文字

写真9　シフトキー併用のメニュー

写真10　メニューメンテ

選択.Xのように，スクロールバーの上でメニューを開くなど，多少「？」な操作体系もありますから，ちょっと様子が変だなと思ったら，積極的に右プレスすることをおすすめします。

SX-WINDOWを使いこんできたり，新しいアプリケーションを手に入れたりしたら，メニューを書き換えたくなってくるものです。そこで利用するものが，メニューメンテです。たとえば，デスクアクセサリアイコンから示すメニュー。デフォルトではアクセサリが登録されていますが，私は「システムの環境を改変する設定メニュー」として定義し直しました（写真10）。

メニューメンテを起動しましょう。メニュー番号2がこのメニューです。メニューメンテの操作方法はマニュアルに譲ります。いかがでしょう？　フォントマン設定にはIFM.Xが，ビデオマン設定にはIVM.Xが実行ファイル名として登録されています。この2つは，1度目の実行は常駐ですが，2度目の実行は環境設定になるからです。この2つだけの特殊ルールなので覚えておいてください。

スタート画面の設定，画面状態の記憶は，別のメニューにあったもので，実行ファイルがない変わったメニューなのですが，これらは項目カット＆ペーストを利用することにより，ほかのシンボルアイコンのメニューに移動できます。

アイコン以外にもウィンドウメニューが利用できます。とはいっても，これはほとんどディレクトリウィンドウ用のメニューの編集ですが。

ディレクトリウィンドウの，ファイルアイコンがなにもないところで出るメニューのいちばん下には「クローズ　？Q」と書いてあります。？に謎のマークがあるはずです。これはショートカットキーといって，メニューを開かなくても，OPT.1＋ある文字でそれを直接実行するという設定です。この設定は，メニューメンテのメニュータイプを「ウィンドウ用」にすると現れます（写真11）。

ごらんのとおり，クローズのところを選択すると，ショートカットキーのところには，Qが表示されているはずです。ショートカットキーは慣れてくると次第に使うようになる便利な機能です。私はショートカットキーをよく使うため，写真12のように，ディレクトリウィンドウ内で使うメニューをこのようにショートカットキーで設定しました。

ショートカットキーを利用するようになると，ウィンドウシステムがますます使いやすくなってきます。残念ながら，ディレクトリウィンドウ以外のウィンドウ用メニューは，システムで管理されておらず，アプリケーションごとに管理されているので，なにかショートカットキーを登録したくてもできないことが多いのが実情です。

アイコンメンテ

アイコンメンテというのは，ほとんどすべてのファイルアイコン，シンボルアイコンなどを，登録したり変更したりできるようにするものです。通常，ファイル，シンボルアイコンが，セレクトされていない状態でアイコンメンテを起動すると，新規ファイルアイコンを作ります。たとえばファイル名を＊.man, 実行ファイルをシャーペン.Xにしてみましょう。これで，manという拡張子のアイコンをダブルクリックすると，シャーペンがそのmanという拡張子のファイルをロードして開くわけです。

設定例を見ると，実行オプションに％がついていることが多いのですが，これはダブルクリックされたファイル名を示します。通常はつけなくても，そのアイコンのファイル名は渡されます。オプション指定などでファイル名の位置が指定されている場合にはファイル名を挿入する位置を指定することができるようになります。

さて，アイコンメンテではメニューを通常時とシフト時の2つ登録できます。写真では通常のファイルと同じように9番のメニューが出るように設定しましたが，場合によっては先に説明したメニューメンテを利用して，新規のメニューを作っても構いません。ポップアップメニューの設定の枠の右下の「編集」を選ぶとメニューメンテが起動されます。

ただこれだけではアイコンが表示されませんから，パターンの枠の中で，アイコンパターンを選ばないといけません。ここでは＊.docと同じくモノクロ157番を利用しました。編集ボタンを押すとパターンエディタが立ち上がり，自分で好きなようにエディットすることができます。ここでエデ

写真11　ショートカットの設定

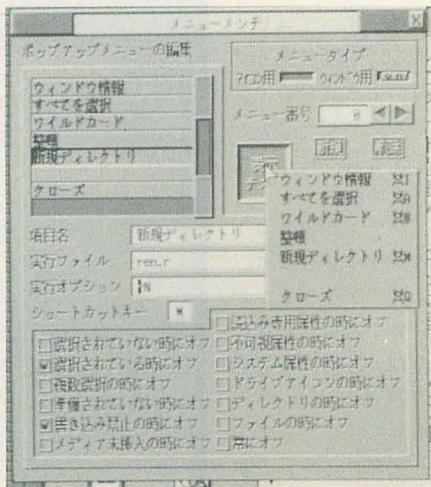



写真12　設定例

ィットすると元のパターンごと変わりますので注意が必要です。パターンエディタに関しては，マニュアルを参照してください。

デスクトップ上の状態の設定で，優先表示すると，システムアイコンなどのように，ウィンドウと重なっても常に優先的に表示するようになります。選択不可にすると，システムアイコンのように，左クリックでも選択できなくなり，ドラッグもできなくなるわけです（ただしOPT.1キーを押しながらなら可能）。

以上で，ファイルアイコンを自由に設定できるようになりました。拡張子ごとの自由な実行設定などが可能になるはずです。

スタートアップメンテ

スタートアップメンテの機能は，起動時の常駐ファイルの設定です。つまりSX-WINDOWのデスクトップの環境設定や機能拡張設定みたいなものですね。コマンド環境に慣れた方なら，AUTOEXEC.BATみたいなものだと考えるのがよいかもしれません。厳密にいえば，SX-WINDOWの大半のソフトウェアはほぼ常駐ソフト（つまりウィンドウが開いている間，常駐しているのと同じ）といえるので，スタートアップで指定するものだけをあえて常駐ソフトと定義するのはおかしいのですが，マニュアルにそういう記述があるのでそれにならうことにします。いい方を変えれば，ウィンドウを開かないのにメモリに居座わるプログラムというのがよいかもしれません。

スタートアップメンテはスタートアップメンテ.Xをダブルクリックして実行します。私はデスクアクセサリメニューに登録したので，ここから呼び出します。

デフォルトではadjust.?，sfile.?，henwin.?，sxcon.?が登録されています。adjustは実画面モードのとき，マウスの移動にあわせてスクロールアウトさせるプログラムです。sfile.?はSX-WINDOWの実行ファイルのパスを学習記録したり，パスを与えたりするもの。henwin.?はSX-WINDOWの中で，日本語変換のためのウィンドウを開くものです。sxcon.?はシャーペンコンソール用のファイルですね。

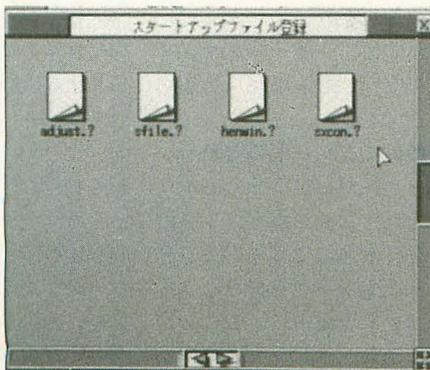



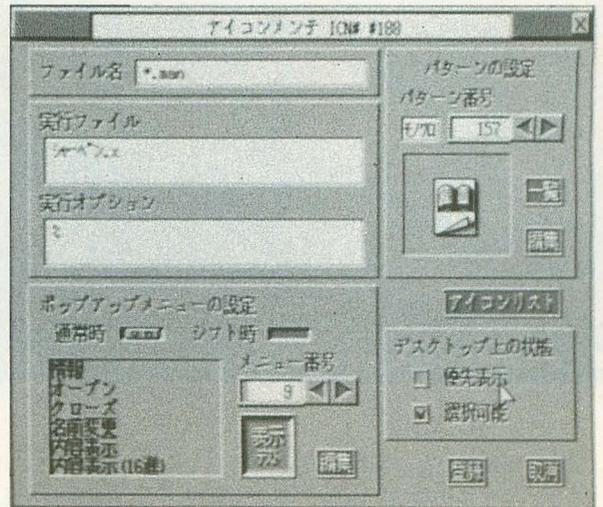
写真14　スタートアップファイルの設定

写真13　アイコンの設定

デフォルトではこれぐらいしか登録されていませんが，シャーペンワープロキットを入れるとSXKEY.Xなどが登録されます。SHELLディレクトリにあるIFM.X，IVM.Xの2つは，フォントマン，ビデオマンです。SX-WINDOWでベクトルフォント（Zeitの書体倶楽部，Z's STAFFのもの），ベジエフォント（ZeitのJGフォント），そして書家万流に同梱されているIFMフォントを利用したいならIFM.Xは必須です。キャンバスで画像を閲覧したりする場合，IVM.Xも必要です。

これらの登録は基本的にスタートアップファイル登録のウィンドウ内にファイルドラッグすることで行えます。

スタートアップファイル登録のウィンドウの中のものを，ダブルクリックすると，オプションの設定ができるようになります。

これ以外に，たとえば「loadlb.r」などをスタートアップファイルに登録すると，builtin.lb，system.lb，icon.lbをデフォルトでメモリにロードするようになります。この3つはSX-WINDOWのリソースファイルといいまして，SX-WINDOW利用中，頻繁にアクセスしにいくものです。これらをメモリ中にロードしておくと，ディスクアクセスを減らし，高速動作するようになるということですね。この登録は，ウィンドウ中のなにもないところでポップアップメニューから新規を選んでやるだけです。

スタートアップメンテの登録は順序がかなり影響します。たとえば，ワープロキットをインストールした場合，SXKEY.X，SXCON.Xの2つはなるべく前のほうに入れておくのがポイントです。この移動は，もちろんドラッグによって行うわけですね。

＊　　＊　　＊

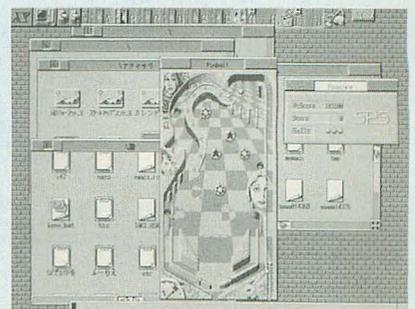
ページの都合上，説明したいところをいろいろ割愛せざるを得ませんでした。もっと初心者向けの記事を！　という要望に応えて，できる限りやさしく書いたつもりでしたが，やっぱりマニュアルは偉大なようです。説明を親切に書きすぎれば，マニュアルより読みづらい文章になるでしょうし。そういうわけで，SX-WINDOWのマニュアルを読みましょうね。うん。

まずはピンボールだ！

ところで，SX-WINDOWをインストールして，たいていの人が最初に行うのはピンボールです。インストールしたドライブはおそらくAドライブになっているはずですから，Aドライブアイコンをダブルクリックし，開いたディレクトリからさらに「アクセサリ」をダブルクリックで開きます。

SX-WINDOWは「ファイル構造」がディレクトリウィンドウを通してそのまま見えるウインドウシステムですから，このあたりはHuman68kのファイル構造をきちっと知らなくてはなりません。さすがにこれをいい始めると長くなるので，このあたりの説明は割愛させていただきます。Human68kのマニュアルを参照してください。

アクセサリには大半のSX上で動くプログラムが入っています。ピンボールもそのなかのひとつです。ディレクトリウィンドウは右の矢印によってスクロールできます。

ピンボール.Xを見つけたらダブルクリック。まずは，100万点台を出すことからSX-WINDOWマスターへの道は開きます。頑張って100万点台を出してください。100万点以上出した方は，証拠写真を添えて「ピンボール100万点係」まで，送ってください。ちなみに，私は出したことがありませんけど。

非SXコマンドを活用する

環境改善の手がかりを探る

Tamura Kento 田村 健人

SX-WINDOWでできなくはないが，コマンドシェルのほうが手軽な処理もある
ツールを片っ端から作り直すのは時間の無駄である
コマンドシェル用のツールをSX-WINDOWの作法で使うための基礎実験を行ってみよう

SX-WINDOWの環境改善の特集ということで，なにか環境を改善する話題を考えてみたがどうも思い浮かばない。これは自分がSX-WINDOWを使っていない証拠……ではなく，現在の環境ではほとんど困っていないということなのだろう。DTPをするわけでもないし，表計算をするわけでもない。音楽を聴き，通信をしながらプログラミングをするぐらいにしか使っていないからかもしれない。猛烈に困っているのはRAMの12Mバイト制限(MS-DOSの640Kバイトと五十歩百歩，むしろひどいくらい)とハードウェアの処理速度だが，これは自分の技術力ではどうしようもない。

3年以上前にSX-WINDOWを常用し始めた当時は，とても環境がよいなどとはいえなかったが，SX-WINDOW自体のバージョンアップ，数々のフリーソフトウェア，また自分自身でプログラムを作ることにより環境を改善してきた。

プログラムを作るのは，欲求があるからである。「不便だから改善したい」「こうしたらもっと便利になるんじゃないだろうか」。欲求が出たら，その実現の可能性を考える。すぐに実現できそうなら，実際にプログラムを作る。あまりにも作業量が多くなり自分だけでは無理そうなら，諦める。で，実現できるかどうかわからないときがいちばん困る。この記事では，どうやって実現したらいいのかいまひとつ不明瞭な欲求について考察してみよう。

コンソールを応用したい

zipやLHAでの圧縮をSX-WINDOW上でやりたい。それならzipやLHAのソースを入手して，ソースを理解して，改造すればよい。しかし，これはかなりの労力を必要とすることが予想できる。まったく異なるプラットフォーム間の移植は大変なのである。

せっかくシャーペンのコンソールがあるのだから，これをなんとか応用して移植の手間を軽減することを考えよう。

コマンドシェル上のプログラムをマルチタスク動作できるようにしているのはSXCON.Xである。シャーペンおよび外部コマンドCONW.EXは入出力の辻褄あわせをしているようだ。実はシャーペンに頼らなくても，SXCON.Xで拡張されるSXコール$a44fによってコマンドシェル上のプログラムを起動できるらしいことは判明している。が，標準入出力などの扱いがまだわからないので，いまのところはシャーペンコンソールに任せることを考えるのが現実的だろう。

あるタスクがほかのタスクを制御する方法は2通り考えられる。

a) ユーザーの指示をエミュレートする

どんなタスクもユーザーに指定された仕事をこなすために存在するわけで，仕事をさせる手段は必ず存在する。「ウィンドウ上にあるボタンを左クリックする」「あるキーを押す」などさまざまである。起動すること自体が重要なこともある。

b) タスク間通信

タスク間通信のインタフェイスを持っているということは，ほかのタスクから制御されることをあらかじめ想定していると考えられる。よって，ユーザーからの指示をエミュレートするより的確な制御ができることがある。

では，シャーペンコンソールにコマンドを実行させる場合，具体的にどうするのかを考えよう。

●エミュレートの実際

ユーザーがコンソールに対して実行したいコマンドを指定する手段は，普通，キーを打ち込むことである。SX-WINDOWでは実際にキーを打たなくてもキーダウンイベントを発生させることができる。ということは，コンソールウィンドウをアクティブにして，コマンドラインを打ち込むよう

リスト1 conexec.c

```
 1: /*
 2:    キーダウンイヴェントを投げつける
 3:    このソースはPDS扱いして結構です
 4:
 5:    conexec.x lha t A:/foo.lzh
 6:    のように起動します。
 7: */
 8: #define _VARLEN           0
 9: #include <task.h>
10: #include <event.h>
11:
12: int main( int argc, char* argv[] ) {
13:     short ide = TSFockB( 0, 0, "ｼｬｰﾍﾟﾝ.X", "¥@-Gｺﾝｿｰﾙ.ENV", NULL, NULL );
14:     int i = 1;
15:     TsEvent er;
16:     TSEventAvail( EM_IDLE, &er );
17:     for ( ;  i < argc;  i ++ ) {
18:    char* p = argv[i];
19:    while ( *p )  EMSet( E_KEYDOWN, (0<<16)+*p++ );
20:    EMSet( E_KEYDOWN, (0<<16)+' ' );
21:     }
22:     EMSet( E_KEYDOWN, (0<<16)+0x0d );
23:                              /* ここで exit<return> をエミュレート
24:                              してもよさそうにみえるが、起動したソフ
25:                              トがキー入力を求めてきた場合にまずい。
26:                                command || exit のようにするとCOMMAND.X
27:                              以外でまずくなる。 */
28: #if 0
29:     for ( ; ; ) {
30:    TSEventAvail( EM_SYSTEM1+EM_SYSTEM2, &er );
31:    if ( er.ts.what2 == ENDTSK )  break;
32:    if ( er.ts.what2 == EXITTSK && (short)er.ts.whom == ide )  break;
33:     }
34: #endif
35:     return 0;
36: }
```

にキーダウンイベントを発生させればよい。

あるタスクが持つウィンドウをアクティブにするにはタスクマネージャイベントのWINDOWSELECTを用いる。しかしシャーペンをシングルウィンドウモードで使用していて，コンソールがウィンドウを持っていないときはWINDOWSELECTを送ってもうまくいかない。また，すでに存在するコンソールがキー入力できる状態であるとは限らない。ということで，コンソールは新たに起動することにする。サンプルをリスト1に示す。

コンソールを起動した直後にコマンドラインを受け取れる状態である保証はないし，発行したキーダウンイベントどおりの文字が入力されるという保証もない。また，実行したコマンドが終了したかどうかを知ることができない。完全ではないが，手軽な方法である。

●タスク間通信の実際

シャーペンにはタスク間通信による制御手段がない。ワープロパック以降のシャーペンではイベント処理を追加できるようになっているので，タスク間通信のインタフェイスを追加する外部コマンドを作ろう。

シャーペンの外部コマンド用ライブラリを使っても「コンソールが入力できる状態にあるか」や「実行したコマンドが終了したか」を知ることはできそうにない。

普通のシャーペンで，OPT1.1＋1を押してから，

conw -C"LHA.x t A:¥foo.lzh"

と入力すると，LHAが実行されてから，そのシャーペンが終了するかもしれない[*1]。これを利用すればなんとかなりそうである。

リスト2がシャーペンをタスク間通信で操作できるようにする外部コマンドである。この外部コマンドcommuni.exを実行しておくと，シャーペンが以下のSX-BASIC形式メッセージを解釈するようになる。

EXMACRO 〈マクロ文字列〉

〈マクロ文字列〉というのはシャーペンのキー定義ファイルに書くのと同じものである。前述のLHAの実行例は，

EXMACRO M1,'conw -C',$22,'LHA.x t A:¥foo.lzh',$22,$0D

となる。このメッセージさえあれば，シャーペンができることなら，なんでもやらせることができる。

以下のような手順で利用する。

0) あらかじめ，シャーペン.ENVのE0マクロでcommuni.exを実行するように設定しておく

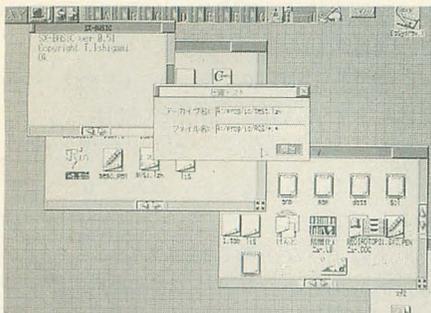
1) “シャーペン.X”というコマンドラインでシャーペンを起動する

2) 起動したタスクにEXMACROメッセージを送る

3) 起動したタスクがいなくなったら，仕事が終了したということである

SX-BASICで簡単なサンプルを作成してみた(リスト3)。アーカイブ名とファイル名を入力して開始ボタンを押すと，LHAで圧縮をする。ファイル名はそのままLHAに

リスト2 communi.c

```
  1: /*
  2:     タスク間通信インタフェイス
  3:     conw の実行が動作するかどうかは環境によります
  4:     すべての .ENV ファイルの E0 で実行するようにしてください
  5:     このソースはPDS扱いして結構です
  6: 
  7: */
  8: #define _VARLEN            0
  9: #include <sxmemory.h>
 10: #include <task.h>
 11: #include <text.h>
 12: #include <mtext.h>
 13: #include <sharpdf2.h>
 14: #include <excom.h>                            /* もみじ狩りPRO */
 15: #include <string.h>
 16: #include <ctype.h>
 17: #define SX_BASIC_SEND      258
 18: 
 19: int job( TEdit** ht, const TsEvent* er, long con ) {/* 0以外で本当の処理をスキップ */
 20:     char* pmes;
 21:     if ( er->ts.what2 != SX_BASIC_SEND )  return 0;
 22:     pmes = *(char**)er->ts.whom;
 23:     if ( 0 == strnicmp( pmes, "EXMACRO ", 8 ) ) {
 24:    long l = strlen( pmes+8 );
 25:    char* pm = MMChPtrNew( l*3/2+1 );          /* ちょっと余裕を持って */
 26:    if ( pm == NULL ) {
 27:        TSErrDialogN( 1, "記憶がない" );
 28:    } else {
 29: #if 0
 30:                                               /* こっちだとうまくいかない */
 31:                                               /* convmacro って違うのかなあ */
 32:        char** hd;
 33:        MMHdlLock( (char**)er->ts.whom );
 34:        if ( convmacro( ht, *(char**)er->ts.whom+8, NULL, 1, 0, &hd ) >= 0 ) {
 35:            MMHdlLock( hd );
 36:            exmacro( ht, 0, MMHdlSizeGet( hd ), *hd );
 37:            MMHdlUnlock( hd );
 38:        } else {
 39:            TSErrDialogN( 1, "コンパイル失敗" );
 40:        }
 41:        MMHdlUnlock( (char**)er->ts.whom );
 42: #else
 43:        char* pr = *(char**)er->ts.whom+8;
 44:        char* pw = pm;
 45:        BOOLEAN err = FALSE;
 46:        while ( *pr && !err ) {
 47:            switch ( *pr ) {
 48:              case '¥'':                          /* 生文字列 */
 49:                pr ++;
 50:                while ( *pr && *pr != '¥'' ) *pw++ = *pr++;
 51:                if ( *pr ) {
 52:                    pr ++;
 53:                } else {
 54:                    TSErrDialogN( 1, "コンパイルエラー ' が閉じていない" );
 55:                    err = TRUE;
 56:                }
 57:                break;
 58:              case 'E':                           /* 3    プレフィックス */
 59:              case 'e':
 60:                *pw++ = 0;  *pw++ = 3;  pr ++;
 61:                break;
 62:              case 'S':                           /* 4    オプション */
 63:              case 's':
 64:                *pw++ = 0;  *pw++ = 4;  pr ++;
 65:                break;
 66:              case '#':                           /* 5    機能番号 */
 67:                *pw++ = 0;  *pw++ = 5;  pr ++;
 68:                break;
 69:              case 'X':                           /* 6    拡張プレフィックス */
 70:              case 'x':
 71:                *pw++ = 0;  *pw++ = 6;  pr ++;
 72:                break;
 73:              case 'M':                           /* 7    外部コマンド */
 74:              case 'm':
 75:                *pw++ = 0;  *pw++ = 7;  pr ++;
 76:                break;
 77:              case 'U':                           /* 8    ユーザ1-2 */
 78:              case 'u':
 79:                *pw++ = 0;  *pw++ = 8;  pr ++;
 80:                break;
 81:              case 'Y':                           /* 9    拡張プレフィックス2 */
 82:              case 'y':
 83:                *pw++ = 0;  *pw++ = 9;  pr ++;
 84:                break;
 85:              case 'Z':                           /* 10   ユーザ3-4 */
 86:              case 'z':
 87:                *pw++ = 0;  *pw++ = 10;  pr ++;
 88:                break;
 89:              case ',':
 90:                pr ++;
 91:                break;
 92:              case '$':                           /* 16進数数値 */
 93:                pr ++;
 94:                {
 95:                    long l = 0;
 96:                    while ( isxdigit( *pr ) ) {
 97:                        l = (l<<4)+(isdigit( *pr ) ? *pr-'0' : (*pr&0x5f)-'A'+10);
 98:                        pr ++;
 99:                    }
100:                    *pw++ = l;
101:                }
102:                break;
103:              default:
104:                if ( isdigit( *pr ) ) {      /* 10進数数値 */
105:                    long l = 0;
106:                    while ( isdigit( *pr ) ) {
107:                        l = l*10+*pr-'0';
108:                        pr ++;
109:                    }
110:                    *pw++ = l;
111:                } else {
112:                    TSErrDialogN( 1, "コンパイルエラー 不正な文字" );
113:                    pr ++;
114:                    err = TRUE;
115:                }
116:                break;
117:            }
118:        }
119:        *pw++ = 0xff;
120:        if ( !err )  exmacro( ht, 0, pw-pm, pm );
121: #endif
122:        MMPtrDispose( pm );
123:    }
124:     } else {
125:    TSErrDialogN( 1, "未知のメッセージが送られてきました" );
126:     }
127:     return 1;
128: }
129: 
130: int main( TEdit** ht, long code, _LASCII param, long idexcom ) {
131:     setevent( ht, (0<<16)+E_SYSTEM1, job, 0 );
132:     setevent( ht, (0<<16)+E_SYSTEM2, job, 0 );
133:     return 0;
134: }
```

LHAを動作させる

渡すので，スペースで区切って複数のファイル名を書いたり，ワイルドカードを利用することができる。

＊1:シャーペンの環境ファイルによっては暴走するようだ。私の環境ではリスト1は期待どおりに動作せず，リスト2は正常に動作する。しかし編集部の環境ではリスト1が正常動作し，リスト2は暴走する。残念ながら原因を究明することができなかったので，打ち込んでも動くかどうかわからないことに留意してほしい。

デスクトップにあるアイコン

デスクトップに置いてあるアイコンの配置にこだわってみたとする。ドライブアイコンなどは頻繁にクリックされてよく位置がずれてしまうので，整頓したい。しかし整頓.Xでは基本的に一列に並べることしかできない。実画面ではなく表示画面を基準に配置を行うのも一長一短だろう。そこで，もっと自由度の高い配置を行うことのできる整頓.Xがほしくなるのである。

まず，整頓.Xがどのような方法でアイコンの移動を行っているのかを理解する必要があるわけだが，現在までに公開されている情報だけではわからないのである。

ウィンドウがウィンドウマネージャに管理されているのと同じように，アイコンがタスクマネージャに管理されているわけではない。アイコンの管理は各タスクが別々に行うようになっており，タスクマネージャはユーティリティコールのみ提供している。しかし整頓.Xが存在するわけであるから，アイコンを移動させるなんらかの手段があるはずである。

ところで，アイコンメンテは選択されているアイコンを起動時の引数にしているようだが，これはどのように実現されているのだろうか。デスクアクセサリアイコンでも選択ファイルがあるかどうかをチェックしてメニューを出している。

アイコンの状態を利用するソフトを作るにしても，アイコンを所持するソフトを作るにしても，必要な情報が正式に公開されていないのが現状なのである。

で，例によってdisる＊2わけである。

ディレクトリ表示ウィンドウdi.rの解析より，選択されたアイコンを得る方法が以下のようになっていることがわかった。

1)　TSPostEventTsk（0，0，0x66，FALSE,FALSE)する

2)　TSEventAvail()して，システムイベント0x67を待つ(なにも選択されていなければ0x67は発生しない)

0x67の内容は以下のとおり。

what2:0x67

whom:セルリストのバイト数

whom2:セルリストへのハンドル

※TSPostEventTsk()ではなく，個別にTSSendMes()してもよい。その場合，TSSendMes()の返事として結果が返ってくる(つまり，TSAnswer2()を使用している)。

＊　　＊　　＊

さらに整頓.Xをdisると，ビルトインコマンドのfallin.rを起動しているだけであるということがわかる。fallin.rはBUILTIN.LBのCODE0010であるので，これを抜き出してまたdisる。ソースをていねいに追って，以下のようなタスク間通信仕様であることがわかった。いずれも，タスクIDが0のSXWIN.Xに送信する。

●デスクトップにあるアイコンを得る

what2:0x0064

TSSendMes()またはTSPostEventTsk?()で送ると以下のような返事が返ってくる。

リスト3 lhatest.sxb

```
 1: /*      SX-BASIC による圧縮サンプル
 2: /*      PDS 扱いしてください
 3: ▼Window Size (272,100),0,0,1,圧縮テスト
 4: /* ここで、初期化に必要な処理を行なって下さい
 5: int whichtext = 0
 6: end
 7: func File_Drop(filename;str)
 8:     if whichtext = 0 then {
 9:         whichtext = 1
10:         Text3.caption = filename
11:     } else {
12:         whichtext = 0
13:         Text4.caption = filename
14:     }
15: endfunc
16: ▼1,Text1 (20,16,100,32),0,0,0,1,3,0,0,0,アーカイヴ名:
17: func Text1_Click()
18: endfunc
19: ▼1,Text2 (32,44,104,56),0,0,0,1,3,0,0,0,ファイル名:
20: func Text2_Click()
21: endfunc
22: ▼1,Text3 (108,14,264,30),0,0,0,0,3,1,0,1,A:/prog/ic/test.lzh
23: func Text3_Click()
24: endfunc
25: func Text3_KeyIn( t;str )
26: endfunc
27: ▼1,Text4 (108,42,264,58),0,0,0,0,3,1,0,1,A:/prog/ic/RCS/*.*
28: func Text4_Click()
29: endfunc
30: func Text4_KeyIn( t;str )
31: endfunc
32: ▼3,StnBtn1 (212,72,260,92),0,0,開始
33: func StnBtn1_Click()
34:     int ide
35:     if Text3.caption = "" or Text4.caption = "" then {
36:         alart( 1, "名前を入れてね" )
37:     } else {
38:         ide = fock( "ｼｬｰﾍﾟﾝ.X" )
39:         if ide < 0 then {
40:             alart( 1, "シャーペンの起動に失敗" )
41:         } else {
42:             sendmes( ide, "EXMACRO M1,'conw -C',34,'lha.x u "+Text3.caption+" "+Text4.ca
ption+"',34,13" )
43:             while findtskn( "ｼｬｰﾍﾟﾝ.X", ide-1 ) > 0
44:             endwhile
45:             alart( 1, "作業完了" )
46:         }
47:     }
48: endfunc
```

無限の拡張性

このシステムで使えそうなコマンドシェル用のコマンドを見てみましょう。システム標準のものではSORTやFIND，WHERE，COPYALLが代表的なものです。それ以外では，普通の人ならば本文にも書かれているように，LHA，ISH，zipといったものが代表となるでしょう。

TeXやZ-MUSICのコンパイルもボタンひとつ，またはシャーペンのセーブと同時にメッセージを送るなどで即座に実行できるようになるでしょうし，そうなるとSX-WINDOW上でできる作業の幅はかなり広がります。

コンソール上ではfishやzshなどのシェルも動作しますので，それなりのスクリプトを書いておけば込み入った処理もできますし，AWKやSEDなどを使えばテキスト処理は完璧にこなせます。シャーペンにうまくリンクすれば，外部での加工をシャーペンの機能のように使うこともできるようになるはずです。いまのところコンソールでの処理完了を知る方法がないので，全自動というわけにはいきませんが，処理待ちの外部ファイルさえ作成すれば完璧になります。

また，本文にあった環境による原因不明の不具合が解決できれば別のコンソールではなく，そのシャーペン自体でコマンドを実行してやることもできるようになるでしょう。

what2:0x0065
whom2:以下のような内容のハンドル
　+0.1　アイコンの数
　+4.1　アイコンの配列へのハンドル

●**アイコンをデスクトップから削除**

what2:0x002c
whom:0
whom2:以下のような内容のハンドル
　+0.1　名前へのポインタ
　+4.b　メディアバイト
　+5.b　ユニット番号
　+6.1　不定値(なんでもよい)

●**アイコンをデスクトップに置く**

what2:0x002d
whom:0
whom2:以下のような内容のハンドル
　+0.1　名前へのポインタ
　+4.b　メディアバイト
　+5.b　ユニット番号
　+6.1　配置する座標

＊　＊　＊

よって，アイコンを移動させたい場合は，

1)　0x0064で存在を確認
2)　0x002cで削除
3)　0x002dで新位置に置く

とすればよいことがわかる。システムアイコン/デスクアクセサリアイコン/ページアイコンは必ず存在するので確認する必要はないだろう。メディアバイト/ユニット番号などについては表1に示す。

デスクトップにあるアイコンの位置などを表示するプログラムicp.x(リスト4)とデスクトップのアイコンを移動させるプログラムicmv.x(リスト5)を作ってみた。icp.xはSXwareだが，標準出力を用いている。コンソール上で実行することを想定しているわけだ。コンソール上で，

A>icp.x -icmv > seiton.bat

として作成したseiton.batをコンソール上で実行すれば，自動的にアイコンの位置を整頓させることができる。

＊　＊　＊

なお，今回掲載のリストはすべて単体では役に立たないと思うので，打ち込まないほうがよい。動作はするはずだが，これらがコンパイルできる環境を持っている人はほとんど存在しないだろう。あくまでも参考である。リストを参考にして使いやすいソフトを作ってほしい。

＊2:disる
動詞(ラ行五段活用)。dis.xで実行ファイルを逆アセンブルし，ソースを得ること。マニアの象徴のような言葉だが，disらなければならない状況というのは一種の寂しさがある。

表1　各アイコンのパラメータ

	media	unit	名前
システムアイコン	0	1	NULL
アクセサリアイコン	0	2	NULL
ページアイコン	0	3	NULL
ドライブアイコン	環境依存	環境依存	ドライブ名：¥
クリーナ	255	0	@：¥
シンボルアイコン	0	ICN#のID＋256	NULL
ファイルアイコン	path依存	path依存	ファイル名

リスト4　icp.c

```
 1: /*
 2:     このソースは PDS 扱いして結構です
 3:     icp -- print placed or selected icons
 4:
 5:  * $Log: icp.c,v $
 6:  * Revision 1.1  1995/04/04  16:41:53  kent
 7:  * Initial revision
 8:  *
 9: */
10: static const char rcsid[] = "$Id: icp.c,v 1.1 1995/04/04 16:41:53 kent Exp kent $";
11: #define _VARLEN          0
12: #include <stdio.h>
13: #include <string.h>
14: #include <task.h>
15: #include <sxmemory.h>
16:
17: typedef struct {
18:     long len;
19:     void** harray;
20: } Icret;
21:
22: int main( int argc, char* argv[] ) {
23:     TsEvent er;
24:     void** h;
25:     long l;
26:     long i = 1;
27:     long _ = 1;
28:     IcState* pis;
29:     long sprim;
30:     int flag = 0;  /* これを1にすると、選択されているファイルの表示になる */
31:     BOOLEAN bat = FALSE;
32:     for ( ;  i < argc;  i ++ ) {
33:    if ( 0 == strcmp( argv[i], "-s" ) )  flag = 1;
34:    if ( 0 == strcmp( argv[i], "-icmv" ) )  bat = TRUE;
35:     }
36:     if ( flag ) {
37:    TSPostEventTsk( 0, 0, 0x0066, 0, 0 );
38:    for (;_;) {
39:        TSEventAvail( EM_IDLE+EM_SYSTEM1+EM_SYSTEM2, &er );
40:        if ( er.ts.what == E_IDLE ) return 1;      /* アイドルイヴェントが来たらあきらめる */
41:        if ( er.ts.what2 == 0x0067 )  break;
42:    }
43:    h = (void**)er.ts.whom2;
44:    sprim = (sizeof(Cell)+sizeof(IcState));
45:    l = er.ts.whom/sprim;
46:     } else {
47:    er.ts.what = E_SYSTEM2;
48:    er.ts.what2 = 0x0064;
49:    TSSendMes( 0, &er );
50:    h = (*(Icret**)er.ts.whom2)->harray;
51:    l = (*(Icret**)er.ts.whom2)->len;
52:    sprim = sizeof(IcState);
53:     }
54:     MMHdlLock( h );
55:     i = 0;
56:     do {
57:    char fn[FILENAME_MAX];
58:    pis = (IcState*)(*h+i*sprim+flag*sizeof(Cell));
59:    TSISRecToStr( pis, fn );
60:    if ( bat == FALSE ) {
61:        fprintf( stdout, "name: %s¥n", fn );
62:        fprintf( stdout, "rect: %d %d  %d %d¥n", pis->bounds.d.left, pis->bounds.d.top, pis-
>bounds.d.right, pis->bounds.d.bottom );
63:        fprintf( stdout, "media: %u, unitno: %u¥n", pis->mediaByte, pis->unitno );
64:    } else {
65:        fprintf( stdout, "icmv.x " );
66:        pis->path ? fprintf( stdout, "%s", fn ) : fprintf( stdout, "-i%u", pis->unitno );
67:        fprintf( stdout, " %d %d¥n", pis->bounds.d.left, pis->bounds.d.top );
68:    }
69:    i ++;
70:     } while ( i < l );
71:     MMHdlUnlock( h );
72:     return 0;
73: }
```

リスト5　icmv.c

```
 1: /*
 2:     このソースはPDS扱いして結構です
 3:
 4:     icmv -- move icons on the root window
 5:
 6:  * $Log: icmv.c,v $
 7:  * Revision 1.1  1995/04/04  16:41:53  kent
 8:  * Initial revision
 9:  *
10: */
11: static const char rcsid[] = "$Id: icmv.c,v 1.1 1995/04/04 16:41:53 kent Exp kent $";
12: #include <stdio.h>
13: #include <stdlib.h>
14: #include <sxmemory.h>
15: #include <event.h>
16: #include <task.h>
17:
18: typedef struct {
19:     char* pname;
20:     unsigned char media;
21:     unsigned char unitno;
22:     Point loc;
23: } IcReq;
24:
25: int main( int argc, char* argv[] ) {
26:     TsEvent er;
27:     IcReq** h;
28:     if ( argc != 4 ) {
29:    fprintf( stderr, "usage: %s <name> <x> <y>¥n", argv[0] );
30:    fprintf( stderr, "¥t<name> := <filename> | -i<unitno>¥n" );
31:    return 1;
32:     }
33:     h = MMChHdlNew( sizeof(IcReq) );
34:     if ( h ) {
35:    IcReq* p = *h;
36:    if ( argv[1][0] == '-' ) {
37:        p->pname = NULL;
38:        p->media = 0;
39:        p->unitno = atoi( argv[1]+2 );
40:    } else {
41:        p->pname = argv[1];
42:        /* きちんと名前があるアイコンの場合、
43:         どうやら mediabyte unitno はでたらめ
44:         でもいいようだ。
45:           しかし $a39a でちゃんと求めておい
46:         たほうが無難ではある */
47:    }
48:
49:    er.ts.what = E_SYSTEM2;
50:    er.ts.when = EMSysTime();
51:    er.ts.what2 = 0x002c;
52:    er.ts.taskID = TSGetID();
53:    er.ts.whom = 0;
54:    er.ts.whom2 = (long)h;
55:    TSSendMes( 0, &er );
56:
57:    p = *h;
58:    p->loc.x_y = LONGWORD( atoi( argv[2] ), atoi( argv[3] ) );
59:    er.ts.what = E_SYSTEM2;
60:    er.ts.when = EMSysTime();
61:    er.ts.what2 = 0x002d;
62:    er.ts.taskID = TSGetID();
63:    er.ts.whom = 0;
64:    er.ts.whom2 = (long)h;
65:    TSSendMes( 0, &er );
66:
67:    MMHdlDispose( h );
68:     }
69:     return 0;
70: }
```

シャーペンの活用

アプリによる環境改善

Nakano Shuichi 中野 修一

システムをいじるだけでなくアプリケーションを使って環境を改善することもできる
ツールのカスタマイズで驚くほど操作環境を変えることもできるのだ
SX-WINDOWでのアプリといえば……やはりシャーペンから始めてみよう

SX-WINDOWの基本システムはけっこうよくできていますので，そのままでもかなり使い勝手のよい環境はできますし，しっかりとカスタマイズしてやれば，たいていのことはできるようになります。

しかし，こういったシステム周りのことばかりいじったり，ファイル操作だけしていてもウィンドウ環境の意味がありませんので，もう少し突っ込んだところを見てみましょう。

ウィンドウ環境で動作しているものは主にアプリケーションです。それらをうまく使ったり，アプリケーション間の連係をうまく取ることでシステムの使い勝手は格段に上がってくることが予想されます。

communi.ex

まずは，けんと君の原稿で作成されたツール，communi.exを使います。残念ながら必ずしもどんな環境でも動くとは限りませんし，今回は実験的な色合いが強いツールですが，せっかくですからできるだけ有効に活用することを考えてみましょう。

環境ファイルによって暴走することがあるということですので，まずは環境ファイルを固定してやりましょう。ここでは標準のコンソール.ENVに例の外部ファイルを登録したものを使用するということに決めます。どうやら，これなら無事に動いていますし。

さて，このツールはどんな局面で威力を発揮するのでしょうか。

テキストファイルの加工などでは，コマンドシェル用にすでにあるツールをすべてSX-WINDOW用に移植するのは馬鹿げています。素直にコンソールを使うというのが正解でしょうが，定形的な処理ならバッチファイルにパラメータを渡すという単純な操作でさえ面倒になることもあります。第一にウィンドウ環境らしくなくなります。

また，SX-WINDOWでタスク間通信をしていて困るのは「いくつかの処理を順々に行いたい」という場合です。普通のSX-WINDOWアプリではほかのタスクの処理が終わったのを確認して次の処理を行う……といった単純な処理がかえって難しくなっているのです。こういったものは，コンソールのCOMMAND.Xが一度にひとつしか処理を実行しないというのもポイントになります。

SX-WINDOWは並列処理は得意ですが，直列処理は得意ではありません。コマンドシェルは並列処理は苦手ですが，直列処理はお手のものです。

こういった性質を利用すれば，「SX-WINDOW用のコマンドをコマンドシェル用に作成する」といったことも考えられるようになってきます。この場合はコマンドシェルをSX-WINDOWのタスク管理ツールとして活用することになるわけです。

本当に定形的な処理はメッセージを送るコマンドさえ作れば，メニューから指定するだけで実行できるようになるでしょう。パラメータを渡したいとか，多少の加工を行って……という場合にはSX-BASICのほうが簡単に処理ができます。

これによって実現される環境を想像してみましょう。ツールはSX-BASICで作成されます。ツールを起動し，アイコンを放り込むと一連の動作が始まる……のようになると思われます。

この方式の欠点は，必ずSX-BASICのウィンドウを開いてしまうことです（何度も苦情はいったのですが……）。

開発時はともかく，アプリが完成した場合，ユーザーが用があるのはウィンドウエンジンのウィンドウだけです。アプリを作る側からいうと，普段は姿を見せずに，ウィンドウエンジンが落とされると一緒に姿を消してほしいものです。エラー発生時だけウィンドウを開いたってそれほど不自然ではありませんし，そんなモードもなくては気軽にツールを作ったりできません。

SX-BASIC自体のウィンドウは画面の隅の邪魔にならないところに置いておきます。アイコン化しておけばいいのでしょうが，なんとなく悪い思い出があるので，私はアイコン化は使っていません。

結局，たくさんのSX-BASICが常駐するのも気持ち悪いので，ひとつのプログラムでできるだけ多くのことをやらせています。AUTOEXEC.BATに相当することはウィンドウの初期化部分（プログラムのメイン部分）に記述すれば簡単にできますし，タイマ関数や各種イベント，メッセージに対応した関数を用意しておいてやれば，ひとつのプログラムでもさまざまな処理ができるようになるのです（マルチタスク性には問題がありますが）。

ひとつの受け口を用意して，そこでいくつかの処理のなかから選択する（あるいは，拡張子などで自動判別する）といった感じになると思います。

イメージがまとまったところで，具体的にこのツールを安全に使用するための環境を構築することを考えてみよう。

もともとのけんと君の構想では，SX-BASICから起動するという方法を取っていました。これはもっとも確実にタスクIDを得る手法です。しかし，いちいちコンソールが開いたり閉じたりするのは美しくありませんので，現在開いているコンソール1個に処理を任せるようにしてみましょう。

この場合問題になるのはタスクIDをどうやって得るかです。findtskn()では起動ファイル名は指定できても，そのコマンドラインまでは指定してサーチできません。SX-BASICでは用意されていないのです。

シャーペンはいくつも立ち上がっている可能性があります。そのなかからコンソールを開いているもの（コマンドラインのパラメータ中に"-Gコンソール.ENV"がある

もの）をみつけなければならないのです。

これにはAラインファンクションを使って起動されているタスクのコマンドラインを獲得し，該当する文字列を探し回るという処理を書かねばなりません。私も半分くらいまではコーディングしたのですが，かなり面倒で，動作中になにかあると非常に危険なプログラムですから，こういったクリティカルな処理をSX-BASICで記述するのはできれば避けたいところです。

ということで，危険なことはしなくても済む方法を探してみましした。もっとも安直な解決法は，リネームしたシャーペンでコンソールを開くという方法です。

シャーペン.X
コンソール.ENV
シャーペン.ARC

を適当な場所にコピーして，それぞれ，

CONSOLE.X
CONSOLE.ENV
CONSOLE.ARC

とリネームします。CONSOLE.ENVはもちろんCOMMUNI.EXを有効にしたものを入れておいてください。

これならば簡単にコンソールのタスクIDを得ることができます。具体的には，

ID＝findtskn("CONSOLE.X",1)

で探せばよいことになります。なお，CONSOLE.Xが起動していないといろいろ都合が悪いので，CONSOLE.Xはスタートアップに登録するなりしておいてください。

希望するコマンドをコンソールで動作させるようにするには，

sendmes(ID,"EXMACRO 'dir',13")

のようにします。あとはこれを織り込んでプログラムを作成するだけですね。ただし，複数の処理を続けてさせる場合，メッセージはある程度間隔をあけて送らないとおかしくなるので注意してください。

今回の活用例には制限があります。

まず，コンソールが独占されていると実行できません。田村氏の基本構想では，いちいち新たにコンソールを開くという手順になっていましたが，それもせわしないので，一度にたくさんは実行しないということにしましょう。

シャーペンの環境

SXアプリの代表例シャーペン。シャーペンをうまく使うことがSX-WINDOWをうまく使うことのいちばんの早道です。

ED.Xのキーアサインしか使っていないとちょっと損をすることもあります。マニュアルはちゃんと読んでみましょう。

変更の手順は，

キー定義の書き出しを行う
できたファイルの中身を書き換える
キー定義の読み込みを行う
キーと環境を保存する

となっています。

●以前に使った検索文字列を呼び出したい

検索コマンドの設定で“-L”をつけるだけです。

●改行コードを置換したい

キーボードからの改行コードの指定には，CTRL-Vの特殊文字入力を使います。ただし，CTRL-MやCTRL-Jだけでは検索できません。必ず，

CTRL-V，CTRL-M，CTRL-V，CTRL-J

の順で入力してください。

そのほか，文字列入力前に改行マークなどをコピーしておき，検索文字にペーストするという手もあります。

●カレント文字列を使う

改行コードをもっと簡単に扱うためにはカレント文字列を利用するのが便利です。これはシャーペンの定義ファイルで検索（sea）や置換（rep）の指定に，-Cをつけるだけです。

ED.Xではカーソル位置からの1ワードがカレント文字列として扱われていましたが，シャーペンでは少し拡張されて，範囲指定された文字列もカレント文字列として扱われます。

たとえば，マウスなどを使って範囲指定をして，デフォルト設定ならESC＋[でカレント文字列の置換モードになります。このとき検索文字列が表示されませんが，構わずリターンキーを押してください。指定範囲に改行マークがあれば，それも置換します。置換文字列側に改行を入れたいときは先ほどと同じ手順を踏むことになりますけど。

ちなみに，改行文字表示はESC＋リターンですね。

●ディレクトリを開く

たまに編集中の文書があるディレクトリを参照したいということがあります。そういった場合は，

#67,'DI.R ',M1,'GETPATH -P1',$0D,$0D ＊ XF3 ＋ O

とすれば，XF3＋Oでその文書のあるディレクトリを開くことができます。

●文書をコピーする

編集中の文書をFDなどにコピーするとき，いちいちディレクトリを開いてマウス操作するのも面倒，ということもあります。

ファイル出力を使って各ドライブのカレントディレクトリににコピーする方法を考えてみましょう。

まず，文書を全選択します。そしてファイル書き出しを実行します。出力する際のファイル名はその文書のファイル名を使いましょう。パス名取り出しコマンドでファイル名だけの取り出しができますので，それを使ってやります。あとはドライブ指定を行えるように，カーソルを先頭に持ってきておけばよいでしょう。

#107,#87,M1,'GETPATH -P2',$0D,#2 ＊ XF3 ＋ C

のようになります。

●CZ系/BJ系プリンタでコード印字

レーザープリンタなどでは半角文字を半角幅で印字してくれるのですが，これらのプリンタではなんの調整も行いません。ということで以前けんと君にsetkind1.exという外部ファイルを作ってもらったのですが（もみじ狩りPRO-68Kに収録），これの具体的な使い方を見てみましょう。

そもそもsetkind1.exは，setkind.exとほぼ同様の機能で，半角文字の属性だけを変えるというツールです。16ドットフォントで作業している場合なら，

#107,M1,'setkind1 -S24,16',$0D,#4 ＊ OPT.1 ＋ P

としておけば，OPT.1＋Pで画面上の半角幅がほぼプリンタのものと同一になり，

#107,M1,'setkind1 -S16,16',$0D,#4 ＊ OPT.1 ＋ @

としておけば，OPT.1＋@で元に戻ります。あとは通常どおりコード印字するだけですね。

setkind1.exの機能は大きさを変えるだけではありませんので，半角文字の書体に凝ったりするのも面白いかもしれません。

＊　＊　＊

最初にやった事項を踏まえていれば「指定範囲をISHファイルとみなして，ファイル出力し，ISHで変換後そのままLHAで解凍処理を行う」といったこともできます。こういった部分ではバッチファイルの手も借りなければなりませんが，2つの環境がクロスオーバーして，なかなか面白いシステム構成になってきました。

その気になればXL/ImageをSX-WINDOWでマルチタスク実行させることも不可能ではありません……結局はメモリの問題がX68000最後の障壁として残るようです。

高解像度表示の実現に向けて

広さは愛だ

Nakano Shuichi 中野 修一

ウィンドウシステムにとってユーザーが受け取る情報量は多いほうがよい
当然のことながら，広い画面ほど操作はやさしい
ソフトウェアでちらつきをなくすシステムとその活用の指針を紹介しよう

現在，私はメガディスプレイ計画により1024×1024ドットの大きさの画面を手にしています。これでもまだ狭いという感じはあるものの，かなり快適な環境を手にしているといっていいでしょう。

普段使っているシステムならなにも問題はないのですが，ひとたび標準状態のSX環境に移るとまず愕然とするのはあまりの画面の狭さです。慣れとは恐ろしいもので，1年くらい前までこの画面モードでSX-WINDOWを使っていたのが，嘘のように感じられます。

SX-WINDOWにとって1024×1024ドットの画面は別次元の快適さをもたらします。SX-WINDOWシステムではちゃんとマルチウィンドウで作業を行えますので，広ければ広いほど人にやさしい環境になります。

しかし，X68000でメガディスプレイを実現するには，本体改造，そしてさらに高解像度に耐えるディスプレイを新調することが必要でした。あえてそこまでやる人というのはあまり多くはないというのが実情でしょう。

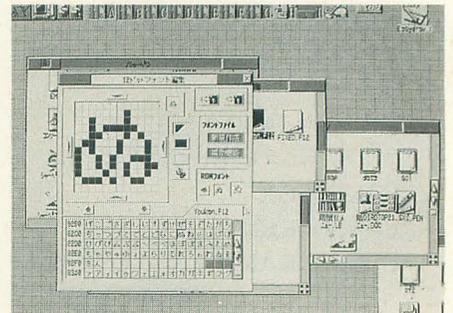
手っとり早くSX-WINDOW環境を改善したいという人がいるなら，まずマシンを買い換えることをおすすめします。10M/16MHz機ならX68030にX68030なら040turboに，そしてディスプレイを換え，最強環境を構築するのです。しかし，元手のかかることゆえ，万人にそうも要求できませんから，ソフトでできる環境改善の可能性を探ってみましょう。

謎のプログラム……

ここに，その昔，石上君の作成したFLICK.Xというプログラムがあります。インタレース走査によって発生するちらつきをなくすツールです。これを使えば通常のモニタで，

SXWIN-G90

などのように1024×848ドットモードにしてもまったく画面がちらつかなくなるという，きわめて強力なツールです。ありがちな話ですが，副作用があります。これを使うと画面が破壊されてしまうのです。

FLICK.Xの原理は簡単で，システムの隙をみてテキストVRAMの偶数ラインを奇数ラインと同じものに換えてやるというものです。実質上，縦方向の解像度は半減します。その結果，文字の大半は判読不能になり，特に12ドットフォントは壊滅的打撃を受けることになります。16ドットフォントもほとんど読めません。24ドットフォントすら危なく，シャーペンなどでアウトラインフォントを使用していれば，使えなくはないかな……という感じの激しいツールでした。

12ドット文字をエディットできる

あまりにしゃれにならないので闇に葬られていたのですが，これをちょっと引っ張り出してみましょう。石上君多忙につき，代打で私が解説します。

美麗12ドット

さて，以前お蔵入りしていたものをどうしていまになって引っ張り出してきたかというと，ちょっとした理由があります。

最近，誌面に出ているSX-WINDOWの画面写真の文字が妙に丸いので不審に思っている人もいたのではないかと思いますが，これは，ピコピコエンジン石田氏の美麗12ドットシステムというものを使って実現したものです。もともと，機械的に作成された12ドット文字の崩れを直そうということで作成されたシステム（およびフォント）

図1-A 通常の画面

```
 */
void e_idle(Comval *pcv)
{
        int        ssp;
        /* ウィンドウを開いている時にしか動作しない */
        if(CMValueGet(pcv->chkHdl)) {
            ssp = SUPER(0);
            {
            int    i, j, k;
            int    *dist,*sour;

            for(i = 0; i < 10; i++) {
                    for(j = 0; j < 4; j++) {
                        sour = (int *)(0xe00000 + 0x20000 * j + 0x80 * pcv->drawY)
                        dist = &sour[32];
                        for(k = 0; k < 32; k++)
                                *dist++ = *sour++;
                    }
                    pcv->drawY = ((pcv->drawY+2) & 0x3ff);
            }
            }
            SUPER(ssp);
        }
}
```

図1-B FLICK.Xにより破壊された画面

ですが，これにはフォントエディタもセットされているので，簡単に自分でフォントを作成することができます。

要するに，これを使えば，ラインを間引いても判読できる文字フォントをSX-WINDOWに登録することができるようになるのです。

まず簡単な実験を行ってみましょう。

すべての文字をエディットするのはあまりに大変なので，とりあえず英数字とカタカナといった半角文字を書き換えてみました。これを美麗12ドット.Rでシステムに登録します。

さらに，高解像度表示でFLICK.Xを起動します。結構，使えそう……かもしれません。具体的にどれくらいの画面になるのかはハードコピーを見てください。実質，6×6ドットのフォントですから，まあこんなもんでしょう。

表示エリアの違いを図2に示します。ちなみにこれはCU-21シリーズを使った場合のものです。

結論

編集部での結論は，ディスプレイがくたびれすぎていて，高解像度での表示は判別しにくいというものでした。FLICKを使わないほうが圧倒的に綺麗な画像が得られますが，ちらつきがひどくて長時間の使用には耐えません。

このシステムを使ったときの作業の様子を図4に示します。ディレクトリウィンドウはなんとか判別可能。シャーペンではアウトラインフォント，Easydrawは太線を使っています。

この状態での画面表示が，全面的に優れているというつもりは毛頭ありません。使いやすいかといわれると，それも困ってしまいます。どうしても通常モードにしなければ作業しにくいことも往々にしてあります。FLICK状態のON/OFFと画面解像度の変更はすぐに行えるので，広い画面が必要なときの選択肢として考えるというのが正解でしょう。

今回は単なる実験として行いましたが，こうなると美麗16ドットくらいはサポートしてもいいかな……というところでしょうか。なお，肝心の美麗12ドットシステム自体は，なにぶん，フォントファイルがないとありがたみが薄いということで，しばらくおあずけです。付録ディスクで供給される予定となっています。いましばらくお待ちください。

図2　作業画面の面積比較

図3　半角文字のみ補正したもの

```
 */
void e_idle(Comval *pcv)
{
        int         ssp;
        if(CMValueGet(pcv->chkHdl)) {
            ssp = SUPER(0);
            {
            int     i, j, k;
            int     *dist,*sour;

            for(i = 0; i < 10; i++) {
                for(j = 0; j < 4; j++) {
                    sour = (int *)(0xc00000 + 0x20000 * j + 0x80 * pcv->drawY)
                    dist = &sour[32];
                    for(k = 0; k < 32; k++)
                        *dist++ = *sour++;
                }
                pcv->drawY = ((pcv->drawY+2) & 0x3ff);
            }
            }
            SUPER(ssp);
        }
}
```

リスト1

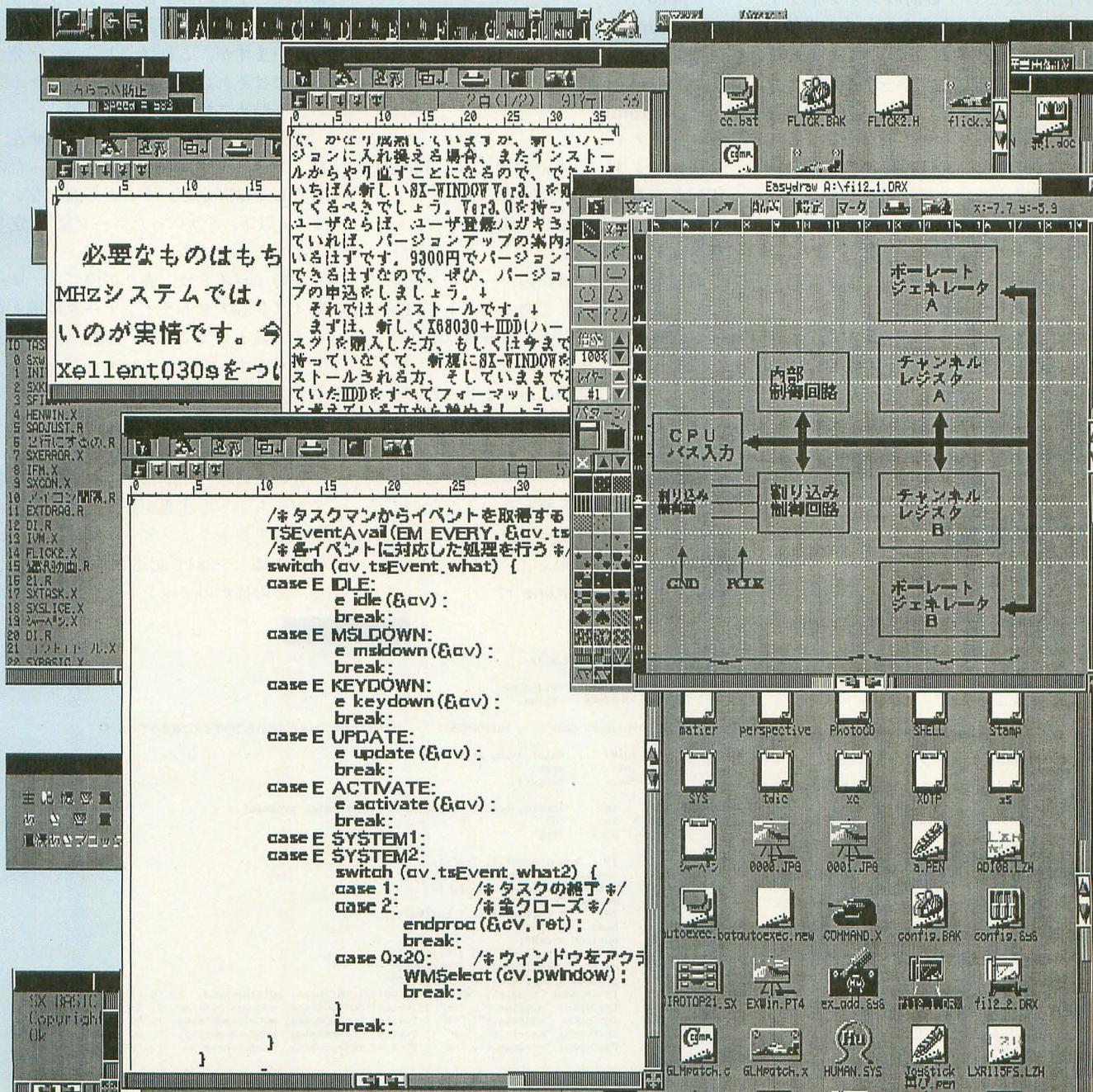
```
 1: /*******************************************************************************
 2:  *      FLICK.c:        /G18モードにおける画面のちらつき防止
 3:  *******************************************************************************
 4:  *
 5:  */
 6: #include <stdio.h>
 7: #include <stdlib.h>
 8: #include <string.h>
 9: #include <sxlib.h>
10: #include "flick2.h"                         /* flick2.cのためのヘッダファイル */
11:
12: char    _sxkernelcomm[] = "sxkernel.x"; /* カーネル起動コマンド */
13:
14: rect     chkrect = { 5, 0, 100, 18 };   /* チェックボックスの座標 */
15:
16: /*******************************************************************************
17:  *      main():         メイン関数
18:  *******************************************************************************
19:  */
20: int main(void)
21: {
22:         Comval cv;
23:         int ret;
24:
25:         /* 変数の初期化 */
26:         cv.pwindow = NULL;
27:         cv.actFlag = FALSE;
28:
29:         /* アプリケーションの初期化 */
30:         if ((ret = init(&cv)) < 0)
31:                 /* 初期化に失敗したので終了する */
32:                 endproc(&cv, ret);
33:
34:         /* イベントループ */
35:         while (1) {
36:                 /* タスクマンからイベントを取得する */
37:                 TSEventAvail(EM_EVERY, &cv.tsEvent);
38:                 /* 各イベントに対応した処理を行う */
39:                 switch (cv.tsEvent.what) {
40:                 case E_IDLE:
41:                         e_idle(&cv);
42:                         break;
43:                 case E_MSLDOWN:
44:                         e_msldown(&cv);
45:                         break;
46:                 case E_KEYDOWN:
47:                         e_keydown(&cv);
48:                         break;
49:                 case E_UPDATE:
50:                         e_update(&cv);
51:                         break;
52:                 case E_ACTIVATE:
53:                         e_activate(&cv);
54:                         break;
```

```
 55:                    case E_SYSTEM1:
 56:                    case E_SYSTEM2:
 57:                            switch (cv.tsEvent.what2) {
 58:                            case 1:         /* タスクの終了 */
 59:                            case 2:         /* 全クローズ */
 60:                                    endproc(&cv, ret);
 61:                                    break;
 62:                            case 0x20:      /* ウィンドウをアクティブにする */
 63:                                    WMSelect(cv.pwindow);
 64:                                    break;
 65:                            }
 66:                            break;
 67:                    }
 68:            }
 69:            return 0;
 70: }
 71:
 72: /****************************************************************************
 73:  *      init():                 初期化処理
 74:  ****************************************************************************
 75:  *      引数:   Comval  *pcv    共通データへのポインタ
 76*  *      戻り値: int             <  0: 初期化失敗(終了)
 77:  *                              >= 0: 初期化成功
 78:  */
 79: int init(Comval *pcv)
 80: {
 81:            task tBuff;
 82:            int ret;
 83:
 84:            /* SXシステムのバージョンを取得する */
 85:            if (SXVer() < SXVER) {
 86:                    /* バージョンが古ければメッセージを表示して終了 */
 87:                    DMError(1, "SXシステムのバージョンが違います。");
 88:                    return -1;
 89:            }
 90:
 91:            /* タスク管理テーブル(tBuff内に返される)を取得する */
 92:            TSGetTdb(&tBuff, -1);
 93:
 94:            /* tbuff.command の解析 */
 95:            ret = TSTakeParam(&tBuff.command, &pcv->wSize, NULL, 0, NULL, NULL);
 96:
 97:            /* ウィンドウの位置指定がある場合 */
 98:            if ((ret & 1) == 0) {
 99:                    /* ウィンドウをオープンする左上座標を得る */
100:                    *(int *)&pcv->wSize = TSGetWindowPos();
101:
102:                    /* ウィンドウの幅指定 */
103:                    pcv->wSize.right  = pcv->wSize.left + 120;
104:                    /* ウィンドウの高さ指定 */
105:                    pcv->wSize.bottom = pcv->wSize.top + 20;
106:            }
107:
108:            /* ウィンドウを作成する */
109:            pcv->pwindow = WMOpen(
110:                    NULL,                   /* Window Pointer    */
111:                    &pcv->wSize,            /* Rectangle Pointer */
112:                    (LASCII *)"\x14ちらつき防止",
113:                    0,                      /* Boolean           */
114:                    WI_STD2 << 4,           /* Resouce ID        */
115:                    (window *)-1,           /* Window Pointer    */
116:                    -1,                     /* Boolean           */
117:                    TSGetID());             /* Task ID           */
118:            if (pcv->pwindow == 0)
119:                    /* ウィンドウが作成できなかったので終了する */
120:                    return -1;
121:
122:            /* ウィンドウを表示する */
123:            WMShow(pcv->pwindow);
124:
125:            /* チェックボックス作成 */
126:            if (CreateControl(pcv) < 0)
127:                    return -1;
128:
129:            pcv->drawY = 0;
130:
131:            /* 初期化が成功したので0を返す */
132:            return 0;
133: }
134:
135: /****************************************************************************
136:  *      e_idle():               マウス左ボタンダウンイベント処理
137:  ****************************************************************************
138:  *      引数:   ComVal  *pcv    共通データへのポインタ
139:  *      注釈:
140:  *      タイトルバー内でマウスの左ボタンが押された場合の処理を行う。
141:  *      この処理でクローズボタンでの終了、ウィンドウの移動が可能となります。
142:  */
143: void e_idle(Comval *pcv)
144: {
145:            int     ssp;
146:            /* ウィンドウを開いている時にしか動作しない */
147:            if(CMValueGet(pcv->chkHdl)) {
148:                ssp = SUPER(0);
149:                {
150:                int i, j, k;
151:                int *dist,*sour;
152:
153:                for(i = 0; i < 10; i++) {
154:                    for(j = 0; j < 4; j++) {
155:                        sour = (int *)(0xe00000 + 0x20000 * j + 0x80 * pcv->drawY);
156:                        dist = &sour[32];
157:                        for(k = 0; k < 32; k++)
158:                            *dist++ = *sour++;
159:                    }
160:                    pcv->drawY = ((pcv->drawY+2) & 0x3ff);
161:                }
162:                }
163:                SUPER(ssp);
164:            }
165: }
166:
167: /****************************************************************************
168:  *      e_msldown():    マウス左ボタンダウンイベント処理
169:  ****************************************************************************
170:  *      引数:   ComVal  *pcv    共通データへのポインタ
171:  *      注釈:
172:  *      タイトルバー内でマウスの左ボタンが押された場合の処理を行う。
173:  *      この処理でクローズボタンでの終了、ウィンドウの移動が可能となります。
174:  */
175: void e_msldown(Comval *pcv)
176: {
177:            int ret;
178:            window *wtemp;
179:
180:            /* イベントが自分のウィンドウのものかを調べる */
181:            if (pcv->pwindow != (window *)pcv->tsEvent.whom)
182:                    return;                 /* 他のウィンドウの場合リターン */
183:
184:            /* イベントを取除く */
185:            TSGetEvent(EM_EVERY, &pcv->tsEvent);
186:            /* 非アクティブなら */
187:            /* OPT.1 keyが押されているかチェック          */
188:            /* 押されていれば、アクティブ時と同様の処理 */
189:            if (!pcv->actFlag && (EHM_OPT1 & ((event *)&pcv->tsEvent)->eHow) == 0) {
190:                    /* ウィンドウをアクティブにする */
191:                    WMSelect(pcv->pwindow);
192:
193:                    /* 指定の座標がウィンドウのどの部分か調べる */
194:                    ret = WMFind(pcv->tsEvent.whom2, &wtemp);
195:                    /* タイトルバー(ドラッグリージョン)部分かな? */
196:                    /* マウスの左ボタンが押されたままになっているか? */
197:                    if (EMLStill() == 0 || ret != W_INDRAG)
198:                            /* タイトルバー以外か、マウスが離されていたらリターン */
199:                            return;
200:            }
201:            /* tsEvent のイベントに対したデフォルトの処理                      */
202:            /* この場合、クローズボタンのコントロールとウィンドウの移動を    */
203:            /* pwindow で   指定したウィンドウに対して行う                    */
204:            ret = SXCallWindM(pcv->pwindow, &pcv->tsEvent);
205:            switch (ret) {
206:            /* クローズボタンを押された */
207:            case W_INCLOSE:                                 /* 終了する */
208:                    endproc(pcv, 0);                        /* 終了処理 */
209:                    break;
210:            case W_ININSIDE:                                /* ウィンドウの内側 */
211:                    CheckControl(pcv);                      /* チェックボックス状態チェック */
212:                    break;
213:            }
214: }
215:
216: /****************************************************************************
217:  *      e_keydown():    キーダウンイベント処理
218:  ****************************************************************************
219:  *      引数:   Comval  *pcv    共通データへのポインタ
220:  *      注釈:   OPT.1+'Q'キーで終了の処理を行う。
221:  */
222: void e_keydown(Comval *pcv)
223: {
224:            /* 自分のウィンドウがアクティブか調べる */
225:            if (pcv->actFlag != TRUE)
226:                    return;                 /* アクティブでない場合リターン */
227:            /* イベントを取除く             */
228:            TSGetEvent(EM_EVERY, &pcv->tsEvent);
229:                                            /* OPT.1キーが押されたか?       */
230:            if (((event *)&pcv->tsEvent)->eHow & EHM_OPT1) != 0)
231:                                            /* 入力キーの大/小文字を区別せずに分岐 */
232:                    switch (pcv->tsEvent.whom & 0xff5f) {
233:                    case 'Q':
234:                            endproc(pcv, 0); /* 終了手続きへ                */
235:                            break;
236:                    }
237: }
238:
239: /****************************************************************************
240:  *      e_activate():   アクティベートイベント処理
241:  ****************************************************************************
242:  *      引数:   ComVal  *pcv    共通データへのポインタ
243:  *      注釈:
244:  *      アクティブ、インアクティブによるアクティブフラグの切り替えと、
245:  *      イベントマスクの切り替えを行う。
246:  */
247: void e_activate(Comval *pcv)
248: {
249:            /* 自分のウィンドウかどうか調べる */
250:            if (pcv->pwindow == (window *)pcv->tsEvent.whom) {
251:                    pcv->actFlag = TRUE;
252:            } else if (pcv->actFlag) {
253:                    pcv->actFlag = FALSE;
254:            }
255: }
256:
257: /****************************************************************************
258:  *      e_update():     アップデート処理
259:  ****************************************************************************
260:  *      引数:   ComVal  *pcv    共通データへのポインタ
261:  *      注釈:
262:  *      アップデート開始とアップデート終了の処理を行う。
263:  *      この処理を行わないと、システムアイコンのメニューが表示されなくなります。
264:  */
265: void e_update(Comval *pcv)
266: {
267:            /* 自分のウィンドウかどうか調べる */
268:            if (pcv->pwindow == (window *)pcv->tsEvent.whom) {
269:                    /* アップデートリージョンをセット */
270:                    /* アップデートの開始 */
271:                    WMUpdate(pcv->pwindow);
272:
273:                    /* コントロールの再描画 */
274:                    GMSetGraph(&pcv->pwindow->wGraph);      /* 自分をカレントグラフにする */
275:                    GMPenMode(0);                           /* ペンモードセット   */
276:                    GMFontMode(0);                          /* FontMode = pset(0) */
277:                    CMDraw(pcv->pwindow);                   /* コントロールを描く */
278:
279:                    /* リージョンを戻す */
280:                    /* アップデートの終了 */
281:                    WMUpdtOver(pcv->pwindow);
282:            }
283: }
284:
285: /****************************************************************************
286:  *      endProc():      終了手続き
287:  ****************************************************************************
288:  *      引数:   ComVal  *pcv    共通データへのポインタ
289:  *               int     code    終了コード
290:  *      注釈:   ウィンドウの廃棄とプログラムの終了を行う。
291:  */
292: void endproc(Comval *pcv, int code)
293: {
294:            /* ウィンドウポインタが0でない(廃棄されていない)とき */
295:            if (pcv->pwindow != 0) {
296:                    /* コントロールを廃棄 */
297:                    CMKill(pcv->pwindow);
298:                    /* ウィンドウを廃棄する */
299:                    WMDispose(pcv->pwindow);
300:            }
301:            /* プログラムを終了する */
302:            exit(code);
303: }
304:
305: /****************************************************************************
306:  *      CreateControl():        コントロールの作成
307:  ****************************************************************************
308:  *      引数:   ComVal  *pcv    共通データへのポインタ
309:  *      戻り値: int             <  0: 作成失敗
310:  *                              >= 0: 作成成功
311:  */
312: int
313: CreateControl(Comval *pcv)
314: {
315:            /* チェックボックスを作成する */
316:            pcv->chkHdl = CMOpen(pcv->pwindow,
317:                    &chkrect,                       /*描画する位置とサイズ  */
318:                    (LASCII *)"\014ちらつき防止",   /*タイトル              */
319:                    -1,                             /*見えるようにする      */
320:                    0,                              /*初期値 = OFF          */
321:                    0,                              /*最小値                */
322:                    1,                              /*最大値                */
323:                    CI_OTNBTN << 4,                 /*チェックボックスのID=2*/
324:                    0);                             /*コントロールレコードのcUserの初期値*/
325:            if (!pcv->chkHdl)
326:                    return -1;
327:            return 0;
328: }
329:
330: /****************************************************************************
331:  *      CheckControl():         コントロールのチェック
332:  ****************************************************************************
```

```
333:  *       引数:   ComVal  *pcv     共通データへのポインタ
334:  */
335: void CheckControl(Comval *pcv)
336: {
337:         int pcode, val;
338:         control **chdl;
339:
340:         GMSetGraph(&pcv->pwindow->wGraph);          /* 自分をカレントグラフにする */
341:         /* 押されているコントロールの強調表示 */
342:         pcode = SXCallCtrlM(pcv->pwindow, &pcv->tsEvent, 0, 0, 0, &chdl);
343:         if (pcode == C_INCHECK) {                   /* パートコードがエラーでない */
344:                 val = CMValueGet(chdl); /* 自分の値を得る      */
345:                 if (val)
346:                         CMValueSet(chdl, 1);    /* 自分の値を1(ON)に変更 */
347:                 else
348:                         CMValueSet(chdl, 0);    /* 自分の値を0(OFF)に変更 */
349:         }
350: }
```

リスト2

```
 1: /*************************************************************************
 2:  *       flick2.h:       flick2.c用ヘッダファイル
 3:  *************************************************************************
 4:  */
 5: /*       頻繁に使用する情報を、構造体にまとめる   */
 6: typedef struct {
 7:         window  *pwindow;                 /* ウィンドウレコードへのポインタ */
 8:         tsevent tsEvent;                  /* イベントレコード             */
 9:         int     actFlag;                  /* アクティブフラグ             */
10:         rect    wSize;                    /* ウィンドウサイズ             */
11:         control **chkHdl;                 /* チェックボックスへのハンドル */
12:         int     drawY;                    /* 処理を施すY座標              */
13: } Comval;
14:
15: #define SXVER   0x0201                    /* SXシステムのバージョン番号 */
16:
17: #define TRUE    1
18: #define FALSE   0
19:
20: /*      x,y座標からLPoint型を作るマクロ          */
21: #define makelpt(x,y)    ((long)(x)<<16|((long)(y)&0x0000ffff))
22:
23: /*      flick.c */
24: int     init(Comval *);
25: void    endproc(Comval *, int);
26: void    e_idle(Comval *);
27: void    e_msldown(Comval *);
28: void    e_keydown(Comval *);
29: void    e_activate(Comval *);
30: void    e_update(Comval *);
31: int     CreateControl(Comval *);          /* チェックボックスの作成 */
32: void    CheckControl(Comval *);           /* チェックボックスの状態チェック */
```

図4　FLICK.X使用システムでの作業の様子

フォント環境の改善へ

IFMのフォーマットを探る

Kikuchi Isao 菊地 功

今になってようやく整備されてきたフォントシステム
もっともっと手軽に，もっともっと使えるフォントができるように
IFMの半角フォント構造を独自に調べてみました

ご存じのように，SX-WINDOWはver.2からIFM形式という自前のフォントファイルを持っていました。しかしこの当時のIFMフォントは半角しかなく，しかも標準でついてきたもの以外サポートされるきざしもありませんでした。いろいろ不便に思っていた方も少なくないでしょう。

そんなことから，Oh!X編集部でIFMの解析が始まりました。まずは丹氏が解析を始め，私がそれを引き継いだのですが，だいたいのフォーマットがわかりましたので，報告してみようと思います。解析の間にXDTPで全角つきのIFMフォントや，フォントエディットできる書家万流が出てしまいましたが，フォーマットがわかるならば，それなりに有効に活用できるでしょう。

ただし，ここで解説するのは，SX-WINDOWに付属されていた半角IFMフォントのみですので，ご了承ください。

IFMの構成

IFMフォントファイルは，以下の5つのパート順で構成されています。

・ヘッダ
・アウトライン
・アドレステーブル
・フォントメトリック
・フォント展開プログラム

フォントメトリックというのは，各フォントの概形をある規則に従って数値化したもので，カーニングしたときにどの程度文字の間隔を詰めるかを決定するものなのですが，IFMフォントではちょっと違っています。本当ならばフォントメトリックと呼ぶべきではないと思うのですが，利用法が同じなので，あえてそう呼ぶことにします。

とりあえずは，IFMフォントを上の5つのパートに分解するプログラムをリスト1に掲載します。このプログラムは，引数で指定されたIFMフォントファイルから5つのパートをそれぞれ，

head.r
outline.r
address.r
metric.r
program.r

というファイル名でカレントディレクトリに抽出します。以下の説明では，TRAD.IFMから抽出されたこれらのファイルのダンプリストを例として使用します。

●ヘッダ (head.r)

識別文字列や，ほかのパートのアドレスなどの情報を記録してあるパートで，128バイトで構成されています。詳細は表1のようになっています。

ヘッダは特に問題はありませんね。フォント名，フォントIDというのが，SX-WINDOWのフォントマネージャで表示される情報です。フォント名はほかのフォントと重複していても構わないようなのですが，フォントIDはすべて違う値になっていなければなりません。

●アウトライン(outline.r)

フォントのアウトライン情報を記録してあるパートで，5つのパートのうち唯一可変長のパートです。詳細は表2のようになっていますが，これは“！”一文字分で，実際にはそれ以降の文字のアウトラインデータも詰めて格納されています。

先頭からいきなりよくわかりません。ただ，すべてのフォントで同じ値だったので，おそらくは固定ということで問題ないでしょう。以下，そういったデータは「固定？」と示してあります。

次の4バイトは，表2の例では，ワークサイズからアウトラインデータ最後までのバイト数ということで，$B9－$10＝$A9となります。フォントマネージャがなにをやっているのかよくわからないのですが，ただ単にアウトラインデータバイト数＋4ということかもしれません。

続いての2バイトはフォントの幅なのですが，ちょっと注意が必要です。あとで説明するアウトラインデータでは，フォントの高さは1～384で記述されています。しかし，そのほかのフォント幅に関する情報で

リスト1

```
 1: /* ifmsplit.c */
 2:
 3: #include   <stdio.h>
 4: #include   <io.h>
 5:
 6: unsigned char      buf[65536];     /* 現行のIFMファイルはどれもこのサイズに収まっている */
 7:
 8: void        main( argc, argv )
 9: int         argc;
10: char        *argv[];
11: {
12:     int     length, head, outline, address, metric, program;
13:     int     id;
14:     FILE    *fp;
15:
16:     fp = fopen( argv[1], "rb" );
17:     length = filelength( fileno( fp ) );
18:     fread( buf, length, 1, fp );
19:     fclose( fp );
20:     id      = buf[0x70]<<8 | buf[0x71];
21:     head    = 0;
22:     outline = 0x80;
23:     address = buf[0x76]<<8 | buf[0x77];
24:     metric  = buf[0x7a]<<8 | buf[0x7b];
25:     program = buf[0x7e]<<8 | buf[0x7f];
26:     fp =fopen( "head.r", "wb" );    fwrite( buf+head, outline-head,  1, fp ); fclose( fp );
27:     fp=fopen( "outline.r", "wb" );fwrite( buf+outline, address-outline, 1, fp ); fclose(fp);
28:     fp=fopen( "address.r", "wb" );fwrite( buf+address, metric-address,  1, fp ); fclose(fp);
29:     fp=fopen( "metric.r", "wb" ); fwrite( buf+metric,  program-metric,  1, fp ); fclose(fp);
30:     fp=fopen( "program.r", "wb" );fwrite( buf+program, length-program,  1, fp ); fclose(fp);
31:
32:     return;
33: }
```

は，なぜか高さが1～1024としたときの数値で記載されています。フォントサイズ1024というと，書体倶楽部を思い出しますが，なにか関係があってのことなのでしょうか？　よくわかりません。

ということで，アウトラインデータのX方向の最大値をmaxxとすると，フォント幅は(maxx+1)*1024/384となります。なぜ1を足すのかって？　そりゃあ，実際にすべてのフォントを調べたらそうなってたからです。明解ですね。

座標展開用ワークサイズというのは，あくまでも私の推測にすぎません。実際に調べたところ，(頂点数＋アウトライン数＋1）×4という値が入っていたので，きっとアウトラインデータがメモリに展開するときには16ビット絶対座標で，エンドマークは32ビットに展開されるんだろうな，ということです。ちなみに間違えて小さな値を入れてしまうと，ほぼ間違いなく暴走します。

いよいよ本命のアウトラインデータです。さきほど述べたように，アウトラインデータは，フォントの高さを384としたデータで記述されています。アウトラインの始点はXYそれぞれ16ビットの絶対座標で，以降は8ビット符号つき整数（正確には－126～126)のベクトルで表され，最後にアウトラインの終了コード$80がきます。

x(16ビット) y(16ビット)
dx(8ビット) dy(8ビット)
dx dy …… dx dy $80

という具合です。このとき，最初と最後の点は一致していなければならないようで，自動的に閉じてくれたりはしません。

ここで例外があります。ベクトルが8ビットで表現できる範囲を超えていた場合は，以下の3つのパターンが存在します。

・パターン1
dx,dyともに範囲を超えたとき
$81 dx(16ビット) dy(16ビット)……

・パターン2
dxだけが範囲を超えたとき
$7F dx(16ビット) dy(8ビット) ……

・パターン3
dyだけが範囲を超えたとき
dx(8ビット) $7F dy(16ビット) ……

このようにしてアウトラインデータをすべて示したら，最後に全体のエンドコード$80がきます。あらかじめアウトラインデータサイズを示しておきながら，わざわざ全体のエンドコードを置くあたり，やはり書体倶楽部を意識しているのでしょうか。

●アドレステーブル（address.r）

各キャラクタのアウトラインデータの開始位置をファイル先頭からのオフセットで表しているパートで，392バイトで構成されています。詳細は表3のようになっています。

固定でもなく，どうしてもわからないデータがありました。「不明」と示してある部分なのですが，これはフォントによってまちまちであるうえに，適当に書き換えても反映されることもなければ，飛ぶこともありませんでした。いったいなんなんでしょう。ほかにも固定らしいのですが，よくわからない部分があります。この辺は，参照しているかどうかも怪しいところです。

最大フォント幅は，例によってフォントの高さを1024としたときの値です。アウト

表1　ヘッダ

```
00000000  49 46 4D 20 49 4E 43 20 46 6F 6E 74 20 4D 61 6E IFM INC Font Man
00000010  61 67 65 72 20 56 65 72 2E 20 31 2E 30 20 43 6F ager Ver. 1.0 Co
00000020  70 79 72 69 67 68 74 20 31 39 39 31 2C 31 39 39 pyright 1991,199
00000030  32 20 49 4E 43 20 43 6F 2E 2C 4C 74 64 2E 1A 00 2 INC Co.,Ltd...
00000040  54 72 61 64 00 00 00 00 00 00 00 00 00 00 00 00 Trad............
00000050  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000060  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000070  00 80 00 00 00 00 7D 7D 00 00 7F 05 00 00 94 25 . ...}}.......%

00000000 ～    識別文字列
00000040 ～    フォント名
00000070 ～    フォントID
00000072 ～    固定(00 00)
00000074 ～    アドレステーブル
00000078 ～    フォントメトリック
0000007C ～    フォント展開プログラム
```

表2　アウトライン（1文字分）

```
00000000  00 02 00 00 00 A9 00 98 00 00 00 00 00 00 00 00
00000010  00 00 01 48 00 1B 00 20 FD 01 FB 05 FB 07 FD 07
00000020  FE 05 FD 09 FE 09 FE 0A FF 13 FF 0D 00 36 01 0C
00000030  01 0D 00 04 02 0A 02 08 01 04 02 06 02 05 02 05
00000040  06 09 05 05 03 01 02 00 03 FF 05 FB 06 F7 02 FB
00000050  02 FB 03 F7 02 F7 02 F6 01 EC 01 F8 00 C6 FF F6
00000060  00 F9 FF F9 00 FB FE F6 FE F7 FD F7 FE FB FE FB
00000070  FA F7 FB FB FD FF FE 00 80 00 1B 01 09 FA 02 FD
00000080  01 F9 03 FE 02 FC 06 FE 06 FF 03 00 08 01 05 03
00000090  07 04 05 07 04 04 01 02 01 08 00 05 FF 03 FF 07
000000A0  FD 05 FA 03 F8 01 FB 00 FB FE F7 FD FB FE FD FE
000000B0  FE F7 FC FA FE FC 00 80 80

00000000 ～    固定?(00 02)
00000002 ～    ワークサイズからアウトラインデータ最後までのバイト数
00000006 ～    フォント幅 (maxx+1)*8/3 切り捨て
00000010 ～    座標展開用ワークサイズ
00000014 ～    アウトライン1の始点x座標(16ビット)
00000016 ～    アウトライン1の始点y座標(16ビット)
00000018 ～    アウトライン1(詳細は後述)
00000078 ～    アウトライン1のエンドマーク($80)
00000079 ～    アウトライン2の始点x座標(16ビット)
0000007B ～    アウトライン2の始点y座標(16ビット)
0000007D ～    アウトライン2(詳細は後述)
000000B7 ～    アウトライン2のエンドマーク($80)
000000B8 ～    全体のエンドマーク($80)
```

ラインデータのパートで示してあるフォント幅の最大の値ですね。

以下はアドレスが“！”から“～”までの94文字分，各4バイトで格納されています。

●フォントメトリック（metric.r）

フォントメトリックに相当するデータを格納してあるパートで，5408バイトで構成されています。通常のフォントメトリックと違うのは，カーニングしたときに詰める幅そのものを数値で持っていることです。IFMフォントでカーニングが利くのは，同じフォントのアルファベット同士だけです。要するにアルファベットの大文字小文字合わせて52文字です。そこで，その程度ならということで，特定のアルファベットが並んだときの詰め幅を総当たりのテーブルで持っているのです。詳細は表4のようになっていますが，これは“A”1文字分のテーブルで，実際にはそれ以降の文字のテーブルも詰めて格納されています。

それぞれ2バイトで詰め幅を表し，先頭から順に，自分の後ろに“A”，“B”，……，“Z”がきた場合の詰め幅を示しています。表4では，3番目のデータが$A8ですから，文字が“AC”と並んでいた場合には，文字間隔を通常よりも$A8だけ狭くするということになります。ちなみにここの詰め幅もフォント高さを1024としたときの値で示されています。

●フォント展開プログラム（program.r）

適当な番地から逆アセンブルすれば，アセンブルリストが取れるようです。ここでは触れませんが，勇気のある方は解析してみてください。どのIFMフォントでも内容は同じようです。

コンバートする

フォーマットがわかったところで，実際にIFMフォントを作ってみましょう。といっても，エディタを作ったりするのはしんどいので，書体倶楽部のフォントからコンバートしてみます。書体倶楽部のフォントなら，御木氏作成のフォントエディタが1992年6月号の創刊10周年記念PRO-68Kに収録されていましたし，それを使用して作成したフォントもあるかもしれませんしね。

書体倶楽部のフォントのフォーマットについては，同号に掲載されていましたが，その号を持っていない方のためにちょっと説明しておきましょう。

書体倶楽部フォントは，アドレステーブルとアウトラインデータで構成されています。アドレステーブルは，先頭2バイトが水準を示し，その後ろから4バイトずつアウトラインデータの先頭位置($450A)からのオフセットを示しています。ただし，オフセットはリトルエンディアンで格納されているため，X680x0ではバイト列を入れ換えるという作業が必要になります。

また，順番はシフトJISコードで若い順ですが，半角や存在しないコードはスキップされています。第一水準ならばシフトJISコードで$817Fから，第二水準なら$989Fから4418文字分のアドレスが格納されているはずです。

アウトラインデータは，XYそれぞれ10ビットの絶対座標で表されています。基本的に4座標10バイトがパックされたかたちになっているのですが，ここでもリトルエンディアンによってワード単位で上位バイトと下位バイトがひっくり返っています。XY座標がともに1023のときはアウトラインのエンドコードを示し，その直後も1023の場合は全体のエンドコードを示します。これだけです。フォントとして必要最低限のデータを冗長に表現したといった感じですね。

あまり美しくはありませんが，リスト2が書体倶楽部からIFMフォントに変換するプログラムです。

使用法は，

```
A>VF2IFM <VF1file>
        [<IFMfile>] [/IDn]
```

となります。VF1fileは書体倶楽部のファイル名，IFMfileは生成するIFMフォント名です。双方とも拡張子は省略でき，IFMfileを省略すると，書体倶楽部のファイル名の拡張子を“IFM”にすげ替えたファイル名となります。

また，フォント名は生成されたIFMフォントファイルのベース名になります。/IDに続いてはフォントIDを指定します。省略すると，256になりますが，必ずほかのフォントにぶつからない値を指定してください。

このプログラムではメモリ上にIFMの

表3 アドレステーブル

```
00000000  04 00 04 FD 01 00 00 09 00 5E 00 21 00 00 00 00 .........^.!....
00000010  00 00 00 80 00 00 01 39 00 00 02 48 00 00 03 71 ... ..9...H...q
00000020  00 00 06 37 00 00 08 03 00 00 0A 3D 00 00 0A 95 ...7.......=....
00000030  00 00 0C 13 00 00 0D B9 00 00 0F 7B 00 00 0F C5 .......ｹ...{...ﾅ
00000040  00 00 10 49 00 00 10 77 00 00 10 99 00 00 10 DF ...I...w.......ﾟ
00000050  00 00 12 F4 00 00 13 B4 00 00 15 A6 00 00 17 EE ...□..ｴ...ｦ....
00000060  00 00 18 55 00 00 1A 41 00 00 1C A2 00 00 1D 4C ...U...A...｢...L
00000070  00 00 20 30 00 00 22 A6 00 00 22 E5 00 00 23 90 .. 0.."ｦ.."...#.
```

00000000 ～　固定? フォント高さ?
00000002 ～　最大フォント幅
00000004 ～　固定?
00000006 ～　不明
00000008 ～　固定? フォント数と開始キャラクタ?
00000010 ～　キャラクタ「!」のアウトラインデータの開始位置
00000014 ～　キャラクタ「"」のアウトラインデータの開始位置
以下同様

表4 フォントメトリック（に相当するもの）（1文字分）

```
00000000  00 00 00 00 00 A8 00 00 00 00 00 00 00 A8 00 00 .....ｨ.......ｨ..
00000010  00 00 00 00 00 00 00 00 00 00 00 00 00 A8 00 00 .............ｨ..
00000020  00 A8 00 00 00 18 00 A8 00 D8 01 50 01 50 00 00 .ｨ.....ｨ.ﾘ.P.P..
00000030  00 F0 00 00 00 00 00 48 00 48 00 30 00 30 00 00 .  ....H.H.0.0..
00000040  00 30 00 00 00 00 00 F0 00 00 00 00 00 00 00 00 .0.....  .......
00000050  00 48 00 48 00 30 00 00 00 00 00 30 00 48 00 D8 .H.H.0.....0.H.ﾘ
00000060  00 D8 00 00 00 90 00 00
```

00000000 ～　後ろに「A」が付いたときの詰め幅
00000002 ～　後ろに「B」が付いたときの詰め幅
以下同様

各パートを生成していき，最後に書き出しています。ただし，フォント展開プログラムのパートは持っていないので，カレントにifmsplit.xで生成されたprogram.rを置いて実行してください。もし置き忘れてしまい，最後に「カレントにprogram.rが必要です」といわれたら，

copy Mincho.ifm＋program.r Mincho2.ifm /b

といったようにして手動で連結しても構いません。

注意が必要なのは，「‾」です。これは，書体倶楽部で全角キーを入れてSHIFT＋'@'キーを押したときのキャラクタではなく，シフトJISコード$814Dのキャラクタが使用されます。これは，半角文字を全角文字に変換するhantozen()関数の仕様ですので，これでは困るという場合には，リスト2の141行に例外処理を入れて対処してください。

フォーマット自体はどちらもさほど難しくはないのですが，面倒なのは詰め幅の生成です。本来はマニュアルで自然な間隔を指定していくのかもしれませんが，52×52＝2704パターンを手作業で指定させると死んでしまいますので，プログラムでやらせています。実際に画面に2つのアルファベットを表示しておいて，ラスタごとにその間隔を調べ，もっとも狭い間隔から詰め幅を求めるという方法をとっています。それでも2704通り調べるにはかなりの時間がかかりますので，覚悟しておいてください。

完成かな，と思ったところで，編集のU氏から注文が2点入りました。ひとつめは「文字と文字の間にある程度の間隔を入れろ」，もうひとつは「記号などの幅はある程度統一しろ」，ということでした。間隔を空けるにはフォントの左右に適当なスペースを入れればいいとして，幅の統一については以下のような分類と指定がありました。

!"',.:;^_`|~	統一する
()[]{}	統一して，それぞれ右/左寄せ
*+-/0123456789<=>	統一する
#$%%&?@¥	統一する

これらの幅の指定はプログラムではマクロWIDTH1～4で定義してあり（フォントの高さを1024としたときの値），もしフォント幅がこの値より小さいときはスペースを入れ，超えるときには横に圧縮するようになっています。また，マクロSPACEで定義されている値は，注文1のフォント両脇に入れるスペース幅を示しています。

図1

```
FILES    = 80
BUFFERS  = 99 1024
VERIFY   = off
BREAK    = On
DIRSCH      = on
KEY   = A:¥KEY.SYS
LASTDRIVE = Z:
*TITLE    = ¥SHELL¥TITLE.SX
USKCG     = ¥SHELL¥USKCG.SYS
DEVICE    = ¥SYS¥fddevice.x
*DEVICE   = ¥SYS¥MODRV.SYS 4:$20*16*1 4:$20*16*253
DEVICE    = ¥sys¥modrv.sys 5:0
DEVICE    = ¥SYS¥RSDRV.SYS
```

IFMの標準となる半角フォントTRADでの印字例。クセの強い文字もさることながら，文字間がまったくとられていないのでかなり読みにくい

図2

```
FILES    = 80
BUFFERS  = 99 1024
VERIFY   = off
BREAK    = On
DIRSCH      = on
KEY      = A:¥KEY.SYS
LASTDRIVE = Z:
*TITLE    = ¥SHELL¥TITLE.SX
USKCG     = ¥SHELL¥USKCG.SYS
DEVICE    = ¥SYS¥fddevice.x
*DEVICE   = ¥SYS¥MODRV.SYS 4:$20*16*1 4:$20*16*253
DEVICE    = ¥sys¥modrv.sys 5:0
DEVICE    = ¥SYS¥RSDRV.SYS
```

新しく加わったSXゴシックの印字例。素直なゴシック体で，半角文字が全角の半分で等幅になっている。これはこれで非常によい。ただ，やはり文字間はとられていない

図3

```
FILES    = 80
BUFFERS  = 99 1024
VERIFY   = off
BREAK    = On
DIRSCH      = on
KEY   = A:¥KEY.SYS
LASTDRIVE = Z:
*TITLE    = ¥SHELL¥TITLE.SX
USKCG     = ¥SHELL¥USKCG.SYS
DEVICE    = ¥SYS¥fddevice.x
*DEVICE   = ¥SYS¥MODRV.SYS 4:$20*16*1 4:$20*16*253
DEVICE    = ¥sys¥modrv.sys 5:0
DEVICE    = ¥SYS¥RSDRV.SYS
```

今回作成したツールで書体倶楽部からコンバートされたフォントの例。印刷物の活字幅に近くなるように設定されている（はずだが，予定よりちょっと幅が広い。謎だ）

最後に

おかしいなあ，シャープの人の話じゃ，書家万流にIFMのフォーマットを含めた全部の資料が入ってるってことだったのに。愚痴をいってもしようがないのですが。さて，書体倶楽部から変換した半角IFMフォントはどういう利用法があるのでしょうか。もちろんいままでどおりに使っても構わないのですが，使い方のひとつを紹介しましょう。

たとえばシャーペンなんかで書体倶楽部のフォントを使うと，半角文字はなぜか無条件にTradになってしまいますよね。そこで，まずvf2ifm.xで半角IFMを生成し，書家万流で元の書体倶楽部と生成した半角IFMをマザーフォントとしたIFMファイルを作る。そうすると，半角と全角で違和感のないフォントができます。U氏はただひたすらこれがやりたかっただけのようです。私はというと，ただEX-Systemでもっとたくさんのフォントを使いたかっただけなのですが。

リスト2

```
  1: #include    <stdlib.h>
  2: #include    <stdio.h>
  3: #include    <graph.h>
  4: #include    <string.h>
  5: #include    <jfctype.h>
  6: #include    <iocslib.h>
  7: #include    <doslib.h>
  8:
  9: void        usage( void );
 10: int         getoutline( FILE *, int, int );
 11: int         setoutline( int, int, int );
 12: void        getpack( FILE *, short *, short * );
 13: int         SJIStoCODE( unsigned short );
 14: int         writeifm( FILE *fp, int );
 15: void        fontmetric( void );
 16: int         draw( int, int, char, unsigned short );
 17: int         space( int k );
 18:
 19: char        head[128]= "IFM INC Font Manager Ver. 1.0 Copyright 1991,1992 INC Co.,Ltd.
¥x1A";
 20: unsigned char      outline[65536];
 21: int         address[392/4];
 22: short       fmetric[5408/2];
 23:
 24: char        vfname[90];
 25: char        ifmname[90];
 26: int         id=256;
 27: int         maxx2;
 28:
 29: char        *Width[] = {
 30:     "!¥"',.:;^_`|~",          /* 統一幅(384) */
 31:     "([{",                    /* 右寄せ(512) */
 32:     ")]}",                    /* 左寄せ(512) */
 33:     "/*+-=0123456789<>",      /* 統一幅(512) */
 34:     "#$%&?@¥¥"                /* 統一幅(768) */
 35: };
 36:
 37: #define     WIDTH1  384
 38: #define     WIDTH2  512
 39: #define     WIDTH3  512
 40: #define     WIDTH4  512
 41: #define     WIDTH5  768
 42: #define     SPACE   64
 43:
 44: void        main( int ac, char *av[] )
 45: {
 46:     int     i, n;
 47:     FILE    *fp;
 48:
 49:     for( i=1; i<ac; i++ ){
 50:             if( av[i][0]=='-' || av[i][0]=='/' ){
 51:                     switch( ((av[i][1]|0x20)<<8)|(av[i][2]|0x20) ){
 52:                     case 'id':id = atoi( &av[i][3] );
 53:                             break;
 54:                     default:usage();
 55:                             return;
 56:                     }
 57:             } else {
 58:                     if( vfname[0]==0 ) strmfe( vfname, av[i], "VF1" );
 59:                     else if( ifmname[0]==0 ) strcpy( ifmname, av[i] );
 60:                     else {
 61:                             usage();
 62:                             return;
 63:                     }
 64:             }
 65:     }
 66:     if( vfname[0]==0 ){
 67:             usage();
 68:             return;
 69:     }
 70:     if( ifmname[0]==0 ) strmfe( vfname, ifmname, "" );
 71:     fp = fopen( vfname, "rb" );
 72:     if( fp==NULL ){
 73:             printf( "%s が見つかりません。¥n", vfname );
 74:             return;
 75:     }
 76:
 77:     memcpy( &head[0x40], ifmname, 48 );
 78:     head[0x70] = id>>8;
 79:     head[0x71] = id;
 80:
 81:     for( i='!', n=0; i<='~'; i++ ){
 82:             printf( "%c", i );
 83:             address[4+i-'!'] = 0x80+n;
 84:             n = getoutline( fp, i, n );
 85:     }
 86:     fclose( fp );
 87:     address[0] = 0x04000000|maxx2;  /* 上位ワードはたぶん固定、下位ワードは最大文字幅 */
 88:     address[1] = 0x01000009;        /* 上位ワードはたぶん固定、下位ワードは不明 */
 89:     address[2] = 0x005E0021;        /* たぶん固定 */
 90:     i = 0x80+n;
 91:     head[0x74] = i>>24;             /* アドレステーブルのアドレス */
 92:     head[0x75] = i>>16;
 93:     head[0x76] = i>>8;
 94:     head[0x77] = i;
 95:     i = 0x80+n+392;
 96:     head[0x78] = i>>24;             /* フォントメトリックのアドレス */
 97:     head[0x79] = i>>16;
 98:     head[0x7A] = i>>8;
 99:     head[0x7B] = i;
100:     i = 0x80+n+392+5408;
101:     head[0x7C] = i>>24;             /* フォント展開プログラムのアドレス */
102:     head[0x7D] = i>>16;
103:     head[0x7E] = i>>8;
104:     head[0x7F] = i;
105:     fontmetric();
106:
107:     printf( "¥nファイル作成中¥n" );
108:     strmfe( ifmname, ifmname, "IFM" );
109:     fp = fopen( ifmname, "wb" );
110:     if( fp==NULL ){
111:             printf( "%s が作れません。¥n" );
112:             return;
113:     }
114:     i = writeifm( fp, n );
115:     if( i==0 ) printf( "%s を作成しました。¥n", ifmname );
116:     else if( i==1 ) printf( "ディスクフルです。¥n" );
117:     fclose( fp );
118: }
119:
120: void        usage()
121: {
122:     printf( "書体倶楽部フォント(VF1)をＳＸフォント(IFM)に変換します。¥n"
123:             "usage :  VF2IFM <VF1file> [<IFMfile>] [option]¥n"
124:             "¥t/IDn¥tフォントIDを指定します。省略時256¥n" );
125: }
126:
127: /* アウトラインデータを書体倶楽部から読み込み、IFMのバッファに書き込む */
128: int         getoutline( FILE *fp, int c, int n )
129: {
130:     int     offset;
131:     int     i, maxx, minx;
132:     short   x[4], y[4], ox, oy, sx, sy;
133:     int     m1, m2, nn, width;
134:     char    flag=1;
135:
136:     nn = n+20;
137:     m1 = 0; /* 頂点数 */
138:     m2 = 0; /* アウトライン数 */
139:     maxx = 0;
140:     minx = 1024;
141:     offset = SJIStoCODE( hantozen( c ) )*4+2;
142:     fseek( fp, offset, 0 );
143:     offset = fgetc( fp )|(fgetc( fp )<<8)|(fgetc( fp )<<16)|(fgetc( fp )<<24);
144:
145:     if( offset==-1 ){
146:             outline[nn++] = 0x80;
147:             outline[nn++] = 0x80;
148:             m2 = 2;
149:             maxx = 256;
150:     } else if( fseek( fp, offset+0x450A, 0 ) ){
151:             outline[nn++] = 0x80;
152:             outline[nn++] = 0x80;
153:             m2 = 2;
154:             maxx = 256;
155:     } else {
156:     for( ;; ){      /* 左端の座標を拾うループ */
157:             getpack( fp, x, y );
158:             for( i=0; i<4; i++ ){
159:                     if( x[i]==1023 && y[i]==1023 ){
160:                             if( flag ) goto quit1;
161:                             flag = 1;
162:                             continue;
163:                     }
164:                     if( minx>x[i] ) minx = x[i];
165:                     if( maxx<x[i] ) maxx = x[i];
166:                     flag = 0;
167:             }
168:     }
169: quit1:
170:     maxx -= minx;
171:
172:     for( i=0; i<5; i++ ) if( strchr( Width[i], c ) ) break;
173:     switch( i ){
174:     case 0: /* 幅384 */
175:             if( maxx<=WIDTH1 ) minx -= (WIDTH1-maxx)/2;
176:             width = WIDTH1;
177:             break;
178:     case 1: /* 幅512右寄せ */
179:             if( maxx<=WIDTH2 ) minx -= WIDTH2-maxx;
180:             width = WIDTH2;
181:             break;
182:     case 2: /* 幅512左寄せ */
183:             width = WIDTH3;
184:             break;
185:     case 3: /* 幅512 */
186:             if( maxx<=WIDTH4 ) minx -= (WIDTH4-maxx)/2;
187:             width = WIDTH4;
188:             break;
189:     case 4: /* 幅768 */
190:             if( maxx<=WIDTH5 ) minx -= (WIDTH5-maxx)/2;
191:             width = WIDTH5;
192:             break;
193:     default:/* 固有幅 */
194:             width = maxx;
195:     }
196:
197:     fseek( fp, offset+0x450A, 0 );
198:     for( ;; ){      /* メインループ */
199:             getpack( fp, x, y );
200:             for( i=0; i<4; i++ ){
201:                     if( x[i]==1023 && y[i]==1023 ){
202:                             if( flag==0 && (sx!=ox || sy!=oy) ){
203:                                     nn = setoutline( sx-ox, sy-oy, nn );
204:                                     m1++;
205:                             }
```

▶画面を15kHzにすると，ハングった。本体が悪いのかと思ったら，X68030とX68000PROの両方でなるので，ディスプレイが悪いらしい。困ったものだ。　蟻馬　章生(21)京都府

```
206:                        outline[nn++] = 0x80;
207:                        m2++;
208:                        if( flag ) goto quit2;
209:                        flag = 1;
210:                        continue;
211:                }
212:                x[i] -= minx;
213:                if( width<maxx ) x[i] = x[i]*width/maxx;
214:                x[i] = (x[i]+SPACE)*3/8;
215:                y[i] = y[i]*3/8;
216:                m1++;
217:                if( flag ){
218:                        sx = x[i];
219:                        sy = y[i];
220:                        outline[nn++] = x[i]>>8;
221:                        outline[nn++] = x[i]&0xFF;
222:                        outline[nn++] = y[i]>>8;
223:                        outline[nn++] = y[i]&0xFF;
224:                        flag = 0;
225:                } else {
226:                        ox = x[i]-ox;
227:                        oy = y[i]-oy;
228:                        nn = setoutline( ox, oy, nn );
229:                }
230:                ox = x[i];
231:                oy = y[i];
232:        }
233:    }
234: quit2:
235:    maxx = width+SPACE*2;
236:    }
237:    if( maxx>maxx2 ) maxx2 = maxx;
238:    outline[n+1] = 2;                          /* たぶん固定 */
239:    outline[n+2] = (nn-n-16)>>24;              /* アウトラインデータ+4バイト */
240:    outline[n+3] = (nn-n-16)>>16;
241:    outline[n+4] = (nn-n-16)>>8;
242:    outline[n+5] = nn-n-16;
243:    outline[n+6] = maxx>>8;                    /* フォント幅 */
244:    outline[n+7] = maxx;
245:    outline[n+16] = ((m1+m2)*4)>>24;           /* ワークサイズ */
246:    outline[n+17] = ((m1+m2)*4)>>16;
247:    outline[n+18] = ((m1+m2)*4)>>8;
248:    outline[n+19] = (m1+m2)*4;
249:    return( nn );
250: }
251:
252: /* IFMのバッファに相対座標を書き込む */
253: int         setoutline( int x, int y, int n )
254: {
255:    if( x>=127 || x<=-127 ){
256:            if( y>=127 || y<=-127 ){
257:                    outline[n++] = 0x81;
258:                    outline[n++] = x>>8;
259:                    outline[n++] = x&0xFF;
260:                    outline[n++] = y>>8;
261:                    outline[n++] = y&0xFF;
262:            } else {
263:                    outline[n++] = 0x7F;
264:                    outline[n++] = x>>8;
265:                    outline[n++] = x&0xFF;
266:                    outline[n++] = y;
267:            }
268:    } else {
269:            if( y>=127 || y<=-127 ){
270:                    outline[n++] = x;
271:                    outline[n++] = 0x7F;
272:                    outline[n++] = y>>8;
273:                    outline[n++] = y&0xFF;
274:            } else {
275:                    outline[n++] = x;
276:                    outline[n++] = y;
277:            }
278:    }
279:    return( n );
280: }
281:
282: /* 書体倶楽部から4座標を読み込む */
283: void        getpack( FILE *fp, short *x, short *y )
284: {
285:    unsigned char   chr;
286:
287:    chr = fgetc( fp );
288:    x[0] = chr>>6;
289:    y[0] = (chr&0x3F)<<4;
290:    chr = fgetc( fp );
291:    x[0] |= chr<<2;
292:    chr = fgetc( fp );
293:    x[1] = chr>>2;
294:    y[1] = (chr&0x3)<<8;
295:    chr = fgetc( fp );
296:    y[0] |= chr>>4;
297:    x[1] |= (chr&0xF)<<6;
298:    chr = fgetc( fp );
299:    x[2] = chr<<2;
300:    chr = fgetc( fp );
301:    y[1] |= chr;
302:    chr = fgetc( fp );
303:    y[2] = chr>>4;
304:    x[3] = (chr&0xF)<<6;
305:    chr = fgetc( fp );
306:    x[2] |= chr>>6;
307:    y[2] |= (chr&0x3F)<<4;
308:    chr = fgetc( fp );
309:    y[3] = chr;
310:    chr = fgetc( fp );
311:    x[3] |= chr>>2;
312:    y[3] |= (chr&0x3)<<8;
313: }
314:
315: int         SJIStoCODE( unsigned short c )
316: {
317:    unsigned char   c1, c2;
318:
319:    c1 = c>>8;
320:    c2 = c&0xFF;
321:    if( c1>0x9F ) c1 -= 0x40;
322:    if( c2>0x7F ) c2--;
323:    c = (c1-0x81)*188+(c2-0x40);
324:    return( c );
325: }
326:
327: /* IFMバッファのファイルへの書き出し */
328: int         writeifm( FILE *fp, int n )
329: {
330:    FILE    *fp2;
331:
332:    if( fwrite( head, 1, 128, fp )<128 ) return( 1 );
333:    if( fwrite( outline, 1, n, fp )<n ) return( 1 );
334:    if( fwrite( address, 4, 392/4, fp )<392/4 ) return( 1 );
335:    if( fwrite( fmetric, 2, 5408/2, fp )<5408/2 ) return( 1 );
336:    fp2 = fopen( "program.r", "rb" );
337:    if( fp2==NULL ){
338:            printf( "カレントに program.r が必要です。¥n" );
339:            return( 2 );
340:    }
341:    if( fread( outline, 1, 1088, fp2 )<1088 ){
342:            printf( "program.r が不正です。¥n" );
343:            return( 2 );
344:    }
345:    if( fwrite( outline, 1, 1088, fp )<1088 ) return( 1 );
346:    return( 0 );
347: }
348:
349: /* フォントメトリック?を求める */
350: void        fontmetric()
351: {
352:    int     i, j, k, l=0;
353:
354:    CRTMOD( 16 );
355:    G_CLR_ON();
356:    line( 0, 384, 767, 384, 14, 0xffff );
357:    for( i='A'; i<='Z'; i++ ){
358:            k = draw( 0, 0, i, 15 )*3/8+1;
359:            for( j='A'; j<='Z'; j++ ){
360:                    draw( k, 0, j, 15 );
361:                    fmetric[l++] = space( k );
362:                    draw( k, 0, j, 0 );
363:            }
364:            for( j='a'; j<='z'; j++ ){
365:                    draw( k, 0, j, 15 );
366:                    fmetric[l++] = space( k );
367:                    draw( k, 0, j, 0 );
368:            }
369:            draw( 0, 0, i, 0 );
370:    }
371:    for( i='a'; i<='z'; i++ ){
372:            k = draw( 0, 0, i, 15 )*3/8+1;
373:            for( j='A'; j<='Z'; j++ ){
374:                    draw( k, 0, j, 15 );
375:                    fmetric[l++] = space( k );
376:                    draw( k, 0, j, 0 );
377:            }
378:            for( j='a'; j<='z'; j++ ){
379:                    draw( k, 0, j, 15 );
380:                    fmetric[l++] = space( k );
381:                    draw( k, 0, j, 0 );
382:            }
383:            draw( 0, 0, i, 0 );
384:    }
385: }
386:
387: /* IFMフォントの描画 */
388: int         draw( int x0, int y0, char c, unsigned short p )
389: {
390:    short   x, y, x1, y1, x2, y2;
391:    int     i, j, k;
392:
393:    i = 0x10/4 + (c-'!');
394:    j = address[i]-0x80;
395:    k = (outline[j+6]<<8)|outline[j+7];
396:    x1 = ((outline[j+20]<<8)|outline[j+21]);
397:    y1 = ((outline[j+22]<<8)|outline[j+23])+y0;
398:    if( x1==(short)0x8080 ) return( k );
399:    x1 += x0;
400:    j += 24;
401:    for (;;) {
402:            if ( outline[j] == 0x80 ) {
403:                    j++;
404:                    if ( outline[j] == 0x80 ) break;
405:                    x1 = ((outline[j]<<8)|outline[j+1])+x0; j+= 2;
406:                    y1 = ((outline[j]<<8)|outline[j+1])+y0; j+= 2;
407:                    continue;
408:            }
409:            if ( outline[j] == 0x81 ) {
410:                    j++;
411:                    x = ((outline[j]<<8)|outline[j+1]); j+= 2;
412:                    y = ((outline[j]<<8)|outline[j+1]); j+= 2;
413:            } else if ( outline[j] == 0x7f ) {
414:                    j++;
415:                    x = ((outline[j]<<8)|outline[j+1]); j+= 2;
416:                    y = (char)outline[j]; j++;
417:            } else {
418:                    x = (char)outline[j]; j++;
419:                    if ( outline[j] == 0x7f ) {
420:                            j++;
421:                            y = ((outline[j]<<8)|outline[j+1]); j+= 2;
422:                    } else {
423:                            y = (char)outline[j]; j++;
424:                    }
425:            }
426:            x2 = x1 + x;
427:            y2 = y1 + y;
428:            line( x1, y1, x2, y2, p, 0xffff );
429:            x1 = x2;
430:            y1 = y2;
431:    }
432:    return( k );
433: }
434:
435: /* フォント間の最小間隔を求める */
436: int         space( int x0 )
437: {
438:    unsigned short  *vp;
439:    int     x1, x2, y, min, ssp;
440:
441:    min = x0;
442:    ssp = SUPER( 0 );
443:    vp = (unsigned short *)0xC00000;
444:    for( y=0; y<384; y++ ){
445:            for( x1=0; x1<x0; x1++ ) if( vp[x0-x1-1] ) break;
446:            for( x2=0; x2<min-x1; x2++ ) if( vp[x0+x2] ) break;
447:            if( x1+x2<min ) min = x1+x2;
448:            vp += 1024;
449:    }
450:    SUPER( ssp );
451:    if( (min-=SPACE*3/4)<0 ) min = 0;
452:    return( min*8/3 );
453: }
```

▶5月号を手に入れる1日前にCD-ROMドライブと21インチモニタを買ったところで，しかも市販のグラフィックツールももっていないので，EX-Systemの発表に大変喜んでおります。なるべく安くお願いします。　横尾　崇(22)大阪府

メニューをより使いやすく

コードリソースに手を出す

Ishigami Tatsuya 石上 達也

SX-WINDOWの基幹をなすポップアップメニュー
そのメニューをより使いやすくするため
メニューのユーザーインタフェイス仕様を変更してみよう

SX-WINDOWはとても簡単にインストールできます。専用のインストーラなどなくてもcopyコマンドの使い方さえ知っていれば問題なくインストールできます。

環境設定もほとんど必要ありません。シャーペンに一部複雑な設定機能がありますが通常はいじる必要はありませんし，それ以外のアプリケーションでも（一部の変なツールを除く），ファイルをハードディスク上に展開すればすべての準備が終了するようなものばかりです。

環境設定の項目数とシステムセットアップの自由度というのはたいていトレードオフの関係にあるのですが，SX-WINDOWは，操作系がたいへんうまく設計されているので，デフォルトの設定を変える必要もほとんどありません。このため，項目数の少なさは，弱点にはならず，操作性とあいまって非常に優れた環境を提供しています。

しかし，いくらSX-WINDOWが上手に設計されているからといっても，100％満足しているわけではなく，自分で手を加えたいところが数箇所見受けられるのもまた事実です。

メニューについて思うこと

右ボタンでメニューを表示。メニュー項目を目で追う。必要な項目を探す。マウスカーソルを合わせる。右ボタンを放す。

メニュー項目が，

カット
コピー
ペースト

くらいなら問題はないのですが，最近は，右ボタンを押しただけで，

前方次検索
後方次検索
前方検索..
後方検索..
全域検索..

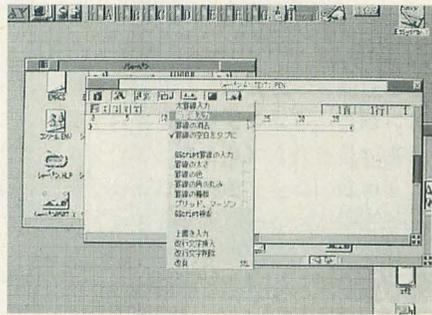



写真1　シャーペンの長いメニュー

⋮

と表示するようなアプリケーションも増えてきました。

項目名を目で追っているうちに，つい，手を放してしまい，あらぬ項目を選んでしまったという経験を持つ人は少なくないでしょう。「取り消し」という機能は，SX-WINDOWのアプリケーションではあまり一般的ではないですから，これが致命的となる場合もあるかもしれません。

右クリックと右プレス。たいていの人は前者2回のほうが後者1回よりも楽に行えると思います。

そこで，今回はこのメニュー操作方法に関していろいろ実験を行いながら，改造を試みます。

最初の右クリックで，メニュー項目の表示。そして，選択したい項目名の上までマウスを移動したところで，2度目のクリック。これで項目確定としましょう。

メニュー操作はSX-WINDOWでも基本をなす操作方法のひとつですが，これをガラリと変更してしまうと戸惑う人もいるかもしれません。従来の方法も残しておきましょう。右ボタンがクリックでなく，プレスされた場合には，従来のように項目確定を受け付けるようにします。

なにをもって，クリックとプレスの区別を行うかは，たいへん難しいのですが，今回はある一定の時間tを決め，

右ボタンが押されている時間＜t
→クリック
右ボタンが押されている時間≧t
→プレス

としました。本来ならtの値はシステムが専用に用意すべきなのかもしれませんが，今回は，ダブルクリック判定時間と共用しています（というわけで，設定は各自SX-WINDOW上のコントロールパネルから行ってください）。

SYSTEM.LB

SX-WINDOWを支える各マネージャの多くは，この「SYSTEM.LB」というファイルにコードリソースとして収められています。参考までに表1がファイルの内容です。MDEFがメニュー，WDEFがウィンドウ，CDEFがコントロールの機能を司るコードリソースです。

SXコールのWMOpen，CMOpenを思い出してください。引数として与えるウィンドウID，コントロールIDの値に対応する機能を実現するためのコードがそれぞれ収められています。

SX-WINDOWに新しい種類のコントロールを追加する場合はCDEF，新しい種類のウィンドウを追加する場合にはWDEFに，それぞれ対応するプログラムを追加します。追加だけでなく，既存のコードリソースを置き換えると，いままでとは違ったSX-WINDOW環境が構築できます。

表1を見るとメニューを実現するコードリソースが2種類あるのがわかります。これは，ID＝0のものがタイトルなしのメニュー（図1），ID＝1のものがタイトルつきのメニュー（図2）を実現するためのものです。

今回は，タイトルつきのメニューとなしのメニュー両方を実現するプログラムを紹介します。一応，追加というかたちをおす

すめしますが，もし，気にいってくだされば，これらの新しい種類のメニューをシャープ純正のものと置き換えることも可能です。ただし，後述するように，今回のプログラムは階層化メニューをサポートしていないので，一部のアプリケーションで不具合が生じます。キャンバスとシャーペンは，なぜかメニューマンを使用せずに自前で階層化メニューを実現しているので，SX-WINDOW ver.3.1のパッケージに入っていたアプリケーションを使っている範囲では不都合は起こりません。

コードコンバータ

いろいろと巷では問題の「開発キットツール集」ですが，その中に「コードコンバータ」と呼ばれるプログラムが入っています。これを使うと，C言語でコードリソースが簡単に書けるようになります。

私は68000のマシン語をほとんど知りません。SX-BASICやFISHの一部に私の書いたアセンブラコードが少し見受けられますが，それらを合計しても，いまだに300行を超えていないと思います。オペコードがあーなってオペランドがこーきてレジスタが云々，というCPUの仕組みそのものはわかるのですが，どうも，レジスタ相対，二重アドレッシング，あたりから頭が回らなくなってしまいます。

そんなわけで，この「コードコンバータ」には，大いに助けられました。詳しいことはよくわからないのですが，なんでもロード時に必要なところを自己書き換えするようなプログラムを先頭に付加するのだそうです。

「SYSTEM.LB」のようなシステムリソースは，いままでマシン語バリバリの人間にしかいじれない，一種の聖域でした。SX-WINDOWの仕組みがわからずに触るとまったく動かなくなる，という点は変わりありませんが，「コードコンバータ」の登場で，私のようにアセンブラはわからなくてもC言語ならわかるという人間も首を突っ込めるようになりました。

テストプログラムの用意

前述のように，新たなメニュー機能を追加するということは，システムリソースである「SYSTEM.LB」を書き換えなければなりません。敷居が若干低くなったとはいえ，まだまだそれなりのリスクを含んだ領域であることには変わりはありません。

これから作るプログラムがうまくSX-WINDOWに組み込めなかった場合，当然のことですが，メニューを表示させようとするとシステムは暴走します。

ですから，今回のプログラムの入力は慎重に行わなければなりません。まず，従来のメニューを書き換える前に，従来のメニューを併用して機能を試してみましょう。つまり，これから作るプログラムをリソースID＝0として登録するのではなく，新たにリソースID＝3として登録するのです。

いまのところリソースID＝3のメニューを使用するプログラムは見あたりませんので今回は3を使います。ほかのプログラムの動作の邪魔にならないように，ID=3のメニューを使用するテストプログラムを作成する必要があります。

新しいメニューがうまく動作するかを試すプログラムをリスト1に示します。実行しても，ウィンドウを開くだけでこのままではなにも行いません。ウィンドウをアクティブにして，ウィンドウ内でマウスの右ボタンを押すことにより，メニューを表示します。どれかの項目を選択すると，ウィンドウ上には，どの項目が選択されたかが，表示されます。なんの変哲もない単純なプログラムですが，新しいメニューの機能をチェックするには十分です。

もし，手もとになんらかのSXプログラムがソースリストごとあれば，そこで使うメニューを交換するだけでリスト1は入力する必要はありません。使用するメニューIDを以下のように3に変更し，コンパイルし

図1　タイトルなしメニュー

図2　タイトルつきメニュー

表1　SYSTEM.LBに含まれるリソース（標準状態）

リソースタイプ	リソースID							
BEEP	0							
PICT	0	1						
CDEF	0	1	2	19	20	64	65	66	67
MDEF	0	1							
WDEF	1	16	32	36	38	39	48	49	50
	51	52	64	99					
DLOG	-4096	-4097	-4098	-4101	-4112	-4113	-4114	-4115	-4116
DITL	-4096	-4097	-4098	-4101	-4112	-4113	-4114	-4115	-4116
CPDF	0	1	2						
PRTM	0								
PAT4	-128	-129							
MENU	-4096								
PRTD	0	1	2	3	4	5			
PRIC	0	1	2	3	4	5			
PALT	1	3	4	-128	-129	-130	16	17	18
PLTL	-11	-12	-13	-14	-22	-23	-24	-33	-34
	-44	-128	-129	-130	3	4			

開発キットのリソースリンカについて

今回のプログラムは単独では動作しません。システムリソースファイル「SYSTEM.LB」に組み込んで，初めて意味を持ちます。ですから，組み込みを行うツールが必要となってきます。今回は，扱うデータがカット＆ペーストでぽいっ，というような種類のものではありませんので，リソースエディタの使用はあきらめ，コマンドライン上からリソースリンカを用いて行うことにしました。

しかし，なぜか，「開発キット」に付属のリソースリンカではうまく動きません。

というわけで，シャープ純正リソースリンカの使用もあきらめ，本誌1994年10月号付録「もみじ狩りPRO-68K」に収録のarlk.xを使用してください。

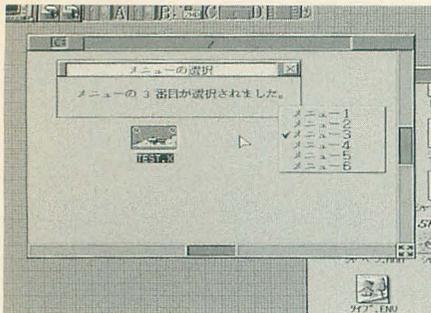



写真2　テストプログラム（チェックマークの表示）

てください。

例)

```
mHdl=MNConvert(NULL,
            "About..",0);
                ↓
mHdl=MNConvert(NULL,
            "About..",3);
```

ただし，暴走してもあまり被害の大きくないプログラムにしておきましょう。

前準備

くどいようですが，これから書き換えようとするファイル「SYSTEM.LB」は，SX-WINDOWの核をなす部分です。いってみれば聖域の一部です。このファイルの破損はすぐさまシステム全体に影響が及びます。必ずバックアップをとり，すぐに復活できるようにしておきましょう。

まず，実験用にディレクトリをひとつ用意してください。容量は約300Kバイト必要です。

そのディレクトリに現在使用中の「SXWIN.X」，「SYSTEM.LB」，「BUILTIN.LB」の3つのファイルをコピーしてきます。これから書き換えるファイルは，「SYSTEM.LB」のみですが，SX-WINDOWは起動時に「SXWIN.X」のあるディレクトリからこれらのファイル検索を行いますので，同じディレクトリに収めておく必要があります。「SXWIN.X」と「BUILTIN.LB」は書き換える必要がないので，書き込み禁止属性にしておくとよいかもしれません。

リスト1，リスト2ともに，無事コンパイルが終わったら，SX-WINDOWを起動し，テストプログラムを実行させます。

右クリックでメニューは出現しますか？選択された項目がウィンドウ上に表示されていますか？　前回選択された項目に，チェックマークが表示されていますか？　右クリックで出っぱなしのメニュー，右プレスで従来どおりのメニュー，ちゃんと表示されていますか？

残念ながら，うまくいかない人は，システムのバージョンや，リストの打ち込み違いがないかを，よくチェックしてください。

ちなみに，リスト2をgccでコンパイルする際，ワーニングは一切出ないようになっています。コンパイル中に，ワーニングメッセージが表示されるようなら，もう一度ソースリストをリスト2と見比べてください。参考までに，リスト2に私の使用したリソースリンカ用インダイレクトファイルを，リスト4にバッチファイルを示します。

インダイレクトファイルにリスト5のものを使うと，リソースID=0(従来のタイトルなしメニュー）に対して今回拡張した機能が組み込まれ，従来のアプリケーションから使用できるようになります。

メニュー定義関数

リソースタイプ「MDEF」のプログラムは，以下のような約束事を守っていれば独自のメニューを実現できるようになっています（このほかにも，プログラムはリロケータブルかつリエントラントである必要がありますが，今回はコードコンバータを使用するので特に問題はありません）。

システムからの引数は，図3のようにスタックに積まれて渡されます。

図中のcommandの値（カッコ内はSXDEF.Hで定義されている値）によって，プログラムは以下の動作を行います（参考文献1より一部引用）。

・command= 0 (CMD_MDRAW)

rectで示される領域（グローバル座標）に，必要な描画を行います。以前表示してあった内容の退避はシステムがすでに行っていますので必要はありません。そのままの状態だと領域内は淡いグレー(G_LGRAY)で塗りつぶされています。グラフポートのセットもシステムがすでに行っています。

・command= 1 (CMD_MSEL)

ptに収められたマウスカーソルの座標(グローバル座標)に従って，メニュー項目の強調表示などを行います。マウスカーソルの座標から算出される値と＊itemの値とが食い違っていた場合には，＊itemに前者を代入します。

・command= 2 (CMD_MCALC)

メニュー表示に必要な四角形の横幅，縦幅を計算し，menuで示されるメニューレコードのmWidth，mHeightに代入します。

参考文献1には，その他の値は拡張用で現在は未使用である，と書かれていますが，SX-WINDOWver.3.0以降では，このほかにcommand=7が送られてくる場合があります。

メニュー項目の選択終了後，点滅動作実

表2　メニュー定義関数の引数

word	command	メニュー定義関数へのコマンドコード
long	Menu	メニューレコードへのハンドル
long	Rect	レクタングルレコードのアドレス
long	pt	ポイント
long	item	アイテム番号を保持するワーク(WORD)のアドレス

返り値はD0.Lで返します。

フリーソフトやウィルスの注意

SX-WINDOWがお手本にしたと思われるMacintoshの世界にも，今回作成したようなプログラムは多数存在します。1994年3月号有田氏の記事（知能機械概論－お茶目な計算機たち－第78回「そしてマウスは運動不足になる」）でも，いろいろなソフトが紹介されていますが，そのほとんどがコードリソース形式のものです。

何度もいうようですが，本来，システムリソースは聖域とでもいうべき，限られた人間にしか踏み込めない領域です。MacintoshのThink Cでは，コードコンバータすら存在せずに，コンパイルオプションスイッチで自動的にコードリソースを吐き出せますから，ある程度の実力のあるユーザーは果敢に挑戦をしていきます。

でもって，もともと聖域であった場所に踏み込んでいったわけですから，もし，不適切なプログラムが紛れ込んでしまった場合は大変です(被害の大きさはX68000でいうところのスタートアップメンテに登録した場合以上，常駐ものの場合以下の事態)。ウイルス性のプログラムはいうに及ばず，ほんの些細なミスでも，その影響はすべてのプログラムに及びます。

極端な例ですが，「フォーマット.X」を開いているときに（フォーマット中ではない）CDEF周りが暴走したとします。ユーザーが実行ボタンを押していなくても，「押した」というイベントが間違って発生してしまうかもしれません。MDEF周りが暴走すれば，「ファイルの保存」などが勝手に実行されてしまわないとも限りません。

今後，システムリソースを書き換えるさまざまなプログラムが登場してくることでしょうが，入手，即，インストールというのは危険です。

本文の前準備で行ったように，いつでもシステムリソースを元に戻せるようにしたうえで，とりあえずテスト環境を作ってみて，問題がないようだったらインストールする，といった慎重な警戒策が必要かもしれません。

行前に1回だけ送られてくる命令で，どうやらメニュー項目の点滅を指示する命令，あるいは階層化メニューの実現のための命令らしいのですが，今回は無視することにしました。解析はマシン語バリバリの田村氏に期待しましょう。

そういうわけで，以上の3つの命令を受け持つ関数を作ってやれば，新しい機能は作れるようになるわけです。

プログラムの概要

このようにして作ったプログラムがリスト2です。いろいろな理由があってコードリソースを初めて作成するような方へのサンプルプログラムとしては，少し不適切なものかもしれません。

コードリソースを作成するときにはmain関数は必要ありません。理由はよくわからないのですが，初めに実行されるべき関数名はCODEMainとします。そのほかにも細かい注意事項はあるのですが，詳しくは「開発キットツール集」のマニュアルを参照してください。

リスト2を見ると，そのCODEMain関数の先頭でいろいろと引数のシフト演算を行っています。これは，システム側が図3のようにパラメータを送ってくるのに対し，C言語では図3-1のように引数を拾ってしまうためです。これは，C言語のパラメータのスタック幅が4バイト固定になっているためです。正しく値を収得するためには，2バイト分ずらして補正を行わなければなりません（図3-2）。

引数補正後は，前述のようにcommandの値によって処理を振り分けてやればよいのですが，今回のプログラムでは以下のような問題点があります。

1) CMD_MSELは，マウスの右ボタンが押されているときにしか呼ばれない。今回のような機能を実現するためには，コードリソース側で2回目の右クリックを待っていなければならない。よって，ここで処理をシステムに戻してしまうわけにはいかない。

2) 今回，右クリックでメニュー表示を行うようにしたが，前述の理由でCMD_MCALC，CMD_MDRAWが終了しないうちに右ボタンが離されてしまうと，CMD_MSELを実行する暇がなくなってしまう。

そこで，commandの値によって以下のように処理を振り分けることにしました（順序はシステムが呼び出す順番）。

・command＝2 (CMD_MCALC)

領域の計算はさほど時間のかかる処理でもありませんので，特に問題はありません。参考文献1に指示のあるとおりに行います。すべての処理に先だって，送られてくる命令ですので，初期化処理が必要な場合はここで行います。

・command＝0 (CMD_MDRAW)

本来ならメニューの描画を行うタイミングです。しかし，10MHz機ではメニューの描画を行っているうちに右クリックが終了してしまう場合があります。今回はなにも行わずに，できるだけ早く制御をシステムに返すようにします。実際の描画はCMD_MSELの最初に行います。

・command＝1 (CMD_MSEL)

本来なら選択動作のみを行う段階ですが，今回のプログラムの場合，前述の理由によりメニューはまっさらです。まずメニュー項目の描画を行います。

そして，マウスカーソルの位置により，システムに何番目の項目が選択されたか知らせるべきなのですが，右ボタンが離されていた場合，そのまま制御が戻ってきません。これでは，出っぱなしのメニューが実現できません。項目が確定するまで処理を独占し，確定した時点で初めてシステムに制御を戻してやります。

プログラムの概要

正直にいって，私はシステムを解析したわけではありませんので，今回のプログラムも，メーカーが意図したような流儀から外れているところがあるかもしれません。参考文献からも情報が得られず，試行錯誤の末，現状に落ち着いたという箇所も少なからずあります。

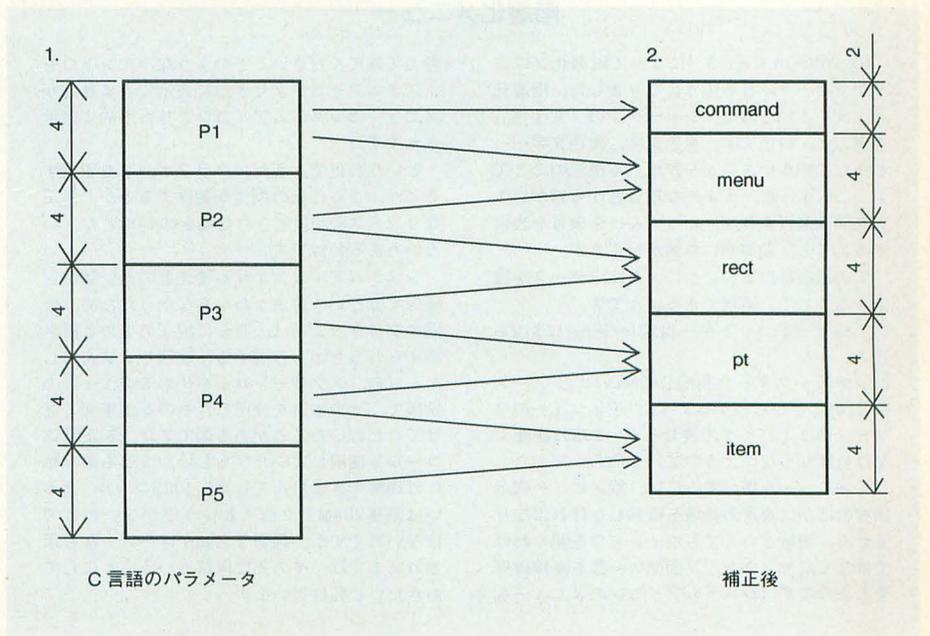
メニューの選択動作は，内部的に2つの処理ルーチンを持っています。従来の動作をするNormalMenu()関数と，今回の目玉であるPopupMenu()関数です。

まず，DoSelect()関数で右ボタンの押されている時間を測定します。測定中に右ボタンが放されればPopupMenu()関数に処理を移します。測定中に，EMDClickGet()関数（システムのダブルクリック時間を返します）よりも時間がたった場合には，自動的にNormalMenu()関数が実行されます。

測定中はマウスカーソルが動かされないとも限りませんので，常に項目名を表示させて，選択/非選択の状態を画面に反映させています。

NormalMenu()関数は，右ボタンが押されているあいだマウスカーソルの動きを見張っています。右ボタンが離されたら，そのとき選択されていた項目の番号を返します。

PopupMenu()関数は，マウスのボタンが再び押されるのをEMRBttn()関数などで見張るだけではうまくいきません。右ボタンを押すと，右ボタンダウンイベントが発生してしまい，そのイベントを拾ったアプリケーションがすぐさまそこに新しいメニューを表示してしまいます。これでは，いつまでたってもメニュー選択を終えることができません。右ボタンクリックを検出したら，すぐにそのイベントを取り除き，ほかのアプリケーションプログラムに検出

図3 CODEMainの引数の拾い方

されないようにしなければなりません。

また，右ボタンの押下を検出し，ただちにメニューを閉じると，別の問題が生じます。どんなに右クリックを素早く行っても，制御をシステムに返された時点では，まだ，右ボタンが押されたままですから，またしてもメニューがそこに表示されてしまいます。右ボタンが完全に離されるまで待ってから制御をシステムに戻します。

細かいこと

●メニュー枠の点滅＆回転

メニュー表示中はその枠を示す部分が点滅/回転しています。これは素直にプログラムを組めばシステムが自動的に行う機能なのですが，今回のプログラムは素直に組んでいませんのでこのサービスが受けられません。

自前で同じことをやらなければいけないのですが，実現法がよくわからなかったため，似たような動作をGMInvertRect関数を用いて行っています。回転部分は，InvertよりもPsetのほうが簡単に実現できるのですが，ラインスタイルつきでグラフィック描画を行えるSX関数はGMInvertRectしか見あたらなかったため，この関数を用いることにしました。メニューの枠が回転して見えるようにするための，ラインスタイルを図4に示します。ただし，GMInvertRect関数はアクセスページを3にしても，なぜか1ページ分しか操作を行わないため，使用色が薄いグレー(G_LGRAY)と濃いグレー(G_DGRAY)となってしまい，純正のものよりもコントラストが落ちてしまいました。

図4：メニューの枠点滅に用いるアルゴリズム

```
・$F0F0   1111   0000   1111   0000
           ↓      ↓      ↓      ↓
          1000   1000   1000   1000
           ↓      ↓      ↓      ↓
・$7878   0111   1000   0111   1000
           ↓      ↓      ↓      ↓
          0100   0100   0100   0100
           ↓      ↓      ↓      ↓
・$3C3C   0011   1100   0011   1100
           ↓      ↓      ↓      ↓
          0010   0010   0010   0010
           ↓      ↓      ↓      ↓
・$1E1E   0001   1110   0001   1110
```

●〼と✓の表示法

表示するメニュー項目はマウスカーソルの位置やイネーブル状態によって色を変えなければなりません。〼や✓のような記号はビットマップで表示させてもよいのですが，できれば文字として扱い，メニュー項目名と一括して処理したいものです。というわけで，「文字選択.X」を使って，漢字コード表を眺めていると，

0xF603に〼
0xF602に✓

が当てがわれていました。SX-WINDOW独自のものなのでコマンドシェル上では表示されないのですが，SX-WINDOW上だとちゃんと扱えます。

●項目名の長さ

メニューの項目名は，LASCII型という，

文字数（1バイト）＋文字列

のような構成の型をとります。普通の項目名の場合は特に問題ないのですが，項目名がない項目（つまり空行のことです）の扱いについて，シャープ製のアプリケーションでも扱いが異なるようです。

最近のアプリケーション（つまり，ver.1.10から追加されたMNConvert関数を用いてメニューレコードを作成しているようなもの，では）

0，0，1，0

となっています（順に，ショートカットなし，チェックマークなし，文字列の長さ1，空文字列）。これに対し，「電卓.X」のように初期のアプリケーションでは，

0，0，0，0

となっています（同，ショートカットなし，チェックマークなし，文字列の長さ0，空文字列）。

さらに，どういう理由からかは知りませんが，メニューアイテムデータは必ず偶数バイト長でなければならない，という決まりがあるので，文字列数が奇数のときは強制的に偶数バイトに調節し，次のメニューアイテムデータを探しにいきます。

以上の理由のために，現在扱っているメニュー項目から次の項目を扱いたい場合には，

```
length=ptr[2] & (~1);
ptr=&ptr[4 + length];
```

のようにポインタを進めます。

●メニュー項目の反転

なんらかの項目を選択すると，メニューが消えるまでに4回ほど項目名が瞬きます。

これは項目確定後，システムから8回ほどCMD_MSELを送ることにより，この機能を実現しているようです（強調表示を4回，通常表示を4回。これを交互に行う）。

行儀のよいプログラムの場合はこれで問題ないのでしょうが，今回のプログラムではあまり都合がよくありません。点滅させようと思ってSX-WINDOWがCMD_MSELを送っても，プログラムは再び右クリックがあるまで制御をシステムには返しません。これでは項目名が瞬いて見えるはずがありませんので同等の機能を自前で実現しなければなりません。

そこでrevCntというフラグを導入しました。通常，この変数の値は0ですが，項目確定後，この変数に8を代入します。メ

階層化メニュー

SX-WINDOWもver.3.1になって階層化メニューがサポートされるようになりました。階層化メニューというのは，シャーペンの「文字色」を選ぶと，白色文字，黒色文字，黄色文字……とさらに細かいメニューが現れる機能のことです。このように，大まかな項目名から細かいものを順次選択すれば，いっぺんに全項目を表示するよりも，効率的に作業が行えます。

この機能は以下のようにメニューデータを設定することで，実現できるようです。

1）　ショートカットキー指定(mSKey)に$1bを代入。

2）　チェックマーク指定(mCheck)にリソースID(このことからわかるように，子メニューのリソースIDは1バイトで表せる0～255の範囲でなければならないようです）を代入。

子メニューを表示するには，親メニュー表示領域のほかに専用の領域を確保しなければなりません。領域といってもウィンドウを開くわけではなく，デスクトップ画面の一部を直接確保するようです（システムアイコンのメニューを考えてみてください。どのようなウィンドウもシステムアイコンより手前に表示できませんが，メニューはシステムアイコンよりも手前に表示されます）。

というわけで，新たにグラフポートを収得するとか，さらに元の画像を確保するとか，不正なリソースIDが設定された場合の処理とか，いろいろあるわけです。

システムアイコンよりも優先度の高い領域を確保するという方法がわからなかったため，今回のプログラムでは，完全に純正のものと置き換えられるものを作成するには至りませんでした（以前，スクリーンセーバーのモジュールの投稿で，その方法を使用したものを編集部で見せていただいたことがあるのですが，未公開SXコールを使用していた（らしい）こともあり私には理解できませんでした）。IOCSコール，あるいは直接VRAMアクセスという手がないわけではないのですが，適切な方法がメーカーから示されるまでは，その先は保留ということにしておきたいと私は思います。

インプログラムでCMD_MSELを受け取ると，実際の処理へ移る前に，この変数の値を調べます。0の場合は，前述のような描画&選択処理を行います。値が0でなければ，

偶数のとき　→　通常表示
奇数のとき　→　反転表示

を行います。その後，revCntの値を1だけ減らし，4回分の点滅処理を行います。

●へこんだ項目名

SX-WINDOWでは，ときどき写真3のようなメニューを見かけます。メニューレコードの内容は図3のようになっており，どこにもこの機能を指示する命令は見あたりません。

私もいろいろ迷ったのですが，その機能は，メニュー項目が空文字列，かつイネーブルフラグが有効，というときに実現される機能のようです（ということは，項目名のないメニューを選択するとスタッフロールが……，という機能を実現できないのですが，まぁいいでしょう）。

●ID＝1のメニューの場合

以上のように，ID＝0相当のメニュー（タイトルなし）を作成してきたわけですが，ID＝1相当のメニュー（タイトルつき）も同様に作成できます。プログラムはリスト1の注釈部分の変更点を書き換えていってください。

まとめ

前々から不満に思っていたメニュー動作の改善を行いました。本当は，ティアオフと呼ばれる機能もつけたかったのですが，それはコードリソースの改造だけでは実現は難しそうです。幸いにも「開発キットツール集」では，「メニューマネージャ」の上を行く「デラックスメニューマネージャ」なるものが供給されていますので，こちらに切り換えるべきなのでしょう。ちなみに，手もとにある最新版のSX-BASICでは，ツールバーやアコーディオンメニューもアイテムとして扱えるようになっているので，システムが自動的に，このティアオフもサポートしてくれています。

いままでは，コンフィグレーションファイルの内容を書き換えるくらいしかカスタマイズのできなかったSX-WINDOWですが，これで，システムの内部まで書き換えられるようになったわけです。

今回と同様にして，MDEFのほかにも，CDEF(コントロール定義関数)，WDEF(ウィンドウ定義関数)，などがC言語で扱えるようになったわけです。もう誰も覚えていないかもしれませんが，「新感覚のスクロールバー」もCDEFの中で定義されているわけです。というわけで，次のターゲットは決まりですね。

SX-WINDOW ver.1.10からはグラフィックの描画ルーチンも書き換えられるようになっていますから，WDEFの書き換えとあわせて面白いことができそうです。

ホレ。やっぱ，グラフィックドライバといえば，ビデオカードですね。

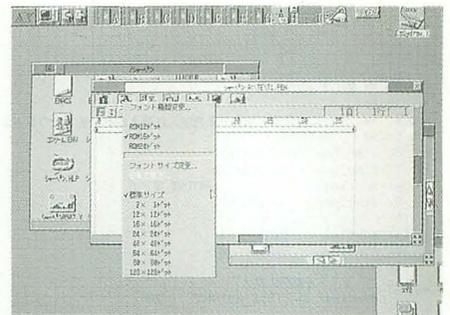
ハードウェアの自作が無理でも，表示をWINDOWSマシンにやらせてしまうという手も考えられますね。描画やウィンドウ操作に関する命令をRS-232Cやジョイスティックポート→パラレルポートでつないで送ってやるわけ。すると，豊富なビデオカードが……。

明後日は入社式だと気がついたところで，皆様，長年のご愛顧本当に有り難うございました。またどこかでお会いしましょう。

写真3　シャーペンのへこみメニュー

●謝辞

今回のプログラムは以下のツールを用いて作成されました。制作者の方々に感謝いたします。

gcc ver 1.00 Tool#1 Based on 1.42
HAS ver 2.25
HLK ver 2.27
CCV ver 1.00
ARLK ver 1.00

参考文献

1）　吉沢正俊「SX-WINDOWプログラミング」ソフトバンク
2）　吉沢正俊「追補版SX-WINDOWプログラミング」ソフトバンク

リスト1　テストプログラム

```
 1: /********************************************************************
 2:  * mitem.c:         メニューのテスト
 3:  ********************************************************************
 4:  * Workroom SX-68K Sample Program (Copyright 1993 SHARP)中,
 5:   サンプル基礎編「MITEM.C」を以下の関数を書き替えてください
 6:  */
 7: /********************************************************************
 8:  * selectMenu():   出たままメニューの作成と選択処理
 9:  ********************************************************************
10:  * 引数:   ComVal *pcv       共通変数へのポインタ
11:  * 戻り値: BOOLEAN          = TRUE:  処理完了
12:  *                           = FALSE: 作成失敗(終了)
13:  */
14: BOOLEAN selectMenu(ComVal *pcv)
15: {
16:     int i, item;
17:     Rect rc;
18:     Menu **menuHdl;                     /* メニューハンドル     */
19:     char str[256];
20: 
21:     static char *menuItem[6] = {      /* メニューアイテム     */
22:             "メニュー1,",
23:             "メニュー2,",
24:             "メニュー3,",
25:             "メニュー4,",
26:             "メニュー5,",
27:             "メニュー6"
28:     };
29: 
30:     /* メニュー文字列の作成 */
31:     str[0] = 0;
32:     for (i = 0; i < 6; i++) {
33:             if (i == pcv->checkMark - 1)
34:                     strcat(str, "!"); /* チェックマークをつける */
35:             strcat(str, menuItem[i]);
36:     }
37:     menuHdl = MNConvert(                /* メニューを作成する         */
38:             NULL,                       /* メニューハンドルを確保する */
39:             str,                        /* メニュー文字列             */
40:             3);                         /* 出たままメニューのリソースID */
41:     if (menuHdl == NULL) {
42:             pcv->errorCode = 6;         /* 作成できなかった           */
43:             return FALSE;               /* 失敗したのでFALSEを返す    */
44:     }
45:     item = MNSelect(                    /* メニューの表示と選択を行う */
46:             menuHdl,                    /* メニューハンドル           */
47:             pcv->event.ev.where.x_y);/*マウスの右ボタンが押された位置*/
48:     MMHdlDispose(menuHdl);              /* メニューハンドルを解放する */
49:     switch (item) {
50:     case 1:                             /* メニュー1          */
51:     case 2:                             /* メニュー2          */
52:     case 3:                             /* メニュー3          */
53:     case 4:                             /* メニュー4          */
54:     case 5:                             /* メニュー5          */
55:     case 6:                             /* メニュー6          */
56:             /* 選択された項目番号にチェックマークをつける */
57:             pcv->checkMark = item;
58:             break;
59:     }
60:     if (pcv->lastMenu != item) {        /* 前回と違う項目が選択されたか? */
61:             pcv->lastMenu = item;       /* 選択された項目番号を保存する */
62:             /* ウィンドウ内全体を書き換える */
63:             rc = pcv->windowPtr->graph.rect;
64:             /* グローバル座標系に変換する */
65:             GMSlideRect(&rc, GMLocalToGlobal(0));
66:             WMAddRect(&rc);             /*アップデートリージョンに追加する*/
67:     }
68:     return TRUE;                        /* 処理が完了したのでTRUEを返す */
69: }
70: 
```

リスト2 MDEF0003.C

```
  1: /*
  2: ** 新しいメニュー
  3: **
  4: **   (//の注釈部を加えるまたは入れ換えるとタイトルつきメニュー)
  5: */
  6: #include    <class.h>
  7: #include    <string.h>
  8: #define             __POINT_T
  9: #include    <sxlib.h>
 10:
 11: #define     TRUE    1
 12: #define     FALSE   0
 13:
 14: /* タイトル部分の高さ */
 15: #define     TITLE_HEIGHT    20
 16:
 17: void        DoDraw(menu **, rect *, point_t);
 18: void        DrawFrame(rect *);
 19: int                 DrawItems(menu **, rect *, point_t);
 20: void        DrawItem(char [], short, rect *, char);
 21: void        DoSelect(menu **, rect *, short *);
 22: void        ReverseItem(menu **, rect *, short *);
 23: int                 NormalMenu(menu **, rect *);
 24: int                 PopupMenu(menu **, rect *);
 25: void        DoCalc(menu **);
 26:
 27: int                 revCnt;         /* 項目点滅用のカウンタ */
 28: short       counter;         /* 枠点滅用のカウンタ */
 29:
 30:
 31: int
 32: /* CODEMain(short command, menu **mhdl, rect *rcWork, point_t pt, short *it
em) */
 33: CODEMain(ULONG p1, ULONG p2, ULONG p3, ULONG p4, ULONG p5)
 34: {
 35:     int         command;
 36:     menu        **mhdl;
 37:     rect        *rcWork;
 38:     point_t pt;
 39:     short       *item;
 40:
 41:     command = p1 >> 16;
 42:     mhdl    = (menu **)((p1 * 0x10000) | (p2 / 0x10000));
 43:     rcWork  = (rect *) ((p2 * 0x10000) | (p3 / 0x10000));
 44:     pt.x_y  =          ((p3 * 0x10000) | (p4 / 0x10000));
 45:     item    = (short *)((p4 * 0x10000) | (p5 / 0x10000));
 46:
 47:     switch(command) {
 48:             case CMD_MDRAW:
 49: /*                  DoDraw(mhdl, rcWork, pt);          */
 50:                     return(TRUE);
 51:             case CMD_MSEL:
 52:                     if(revCnt > 0) {
 53:                             /* 点滅表示 */
 54:                             ReverseItem(mhdl, rcWork, item);
 55:                             revCnt--;
 56:                     } else {
 57:                             /* 10MHz機では、CMD_MDRAWでは間に合わない */
 58:                             /* ので、ここで描画しを行う             */
 59:                             DoDraw(mhdl, rcWork, pt);
 60:                             /* 通常の選択 */
 61:                             DoSelect(mhdl, rcWork, item);
 62:                     }
 63:                     return(TRUE);
 64:             case CMD_MCALC:
 65:                     DoCalc(mhdl);
 66:                     return(TRUE);
 67: #if 0
 68:             default:
 69:                     {
 70:                             char    buff[100];
 71:                             sprintf(buff, "%d mhdl=%x (%d,%d)-(%d,%d) (%d,%d)
item = %d",
 72:                                     command, mhdl, rcWork->left, rcWork->top,
rcWork->right, Work->bottom,
 73:                                     pt.p.x, pt.p.y, *item);
 74:                             DMError(0xff01, buff);
 75:                     }
 76:                     break;
 77: #endif
 78:     }
 79:     return(-1);
 80: }
 81:
 82: void
 83: DoDraw(menu **mhdl, rect *rcWork, point_t pt)
 84: {
 85:     short   oldpage;
 86:     short   oldpenmode;
 87:     short   oldfontmode;
 88:     short   oldforecolor, oldbackcolor;
 89:     point_t oldpensize;
 90:
 91:     /* グラフィック情報の保存 */
 92:     oldpage = GMAPage(3);
 93:     oldpensize.p.x = 1;
 94:     oldpensize.p.y = 1;
 95:     oldpensize.x_y = GMPenSize(oldpensize);
 96:     oldpenmode     = GMPenMode(G_FORE | G_PSET);
 97:
 98:     oldfontmode     = GMFontMode(G_PSET);
 99:     oldforecolor    = GMForeColor(G_LGRAY);
100:     oldbackcolor    = GMBackColor(G_LGRAY);
101:
102:     GMInvertRect(rcWork, 0x0f0f0);
103:
104:     DrawFrame(rcWork);
105:     DrawItems(mhdl, rcWork, pt);
106:
107:     /* グラフィック情報を元に戻す */
108:     GMAPage(oldpage);
109:     GMPenSize(oldpensize);
110:     GMPenMode(oldpenmode);
111:
112:     GMFontMode(oldfontmode);
113:     GMForeColor(oldforecolor);
114:     GMBackColor(oldbackcolor);
115: }
116:
117: /*
118: ** 枠を描く
119: */
120: void
121: DrawFrame(rect *frame)
122: {
123:     point_t loc;
124:     rect    rcWork;
125:
126:     rcWork.left   = frame->left + 2;
127:     rcWork.top    = frame->top + 1;
128:     rcWork.right  = frame->right - 2;
129:     rcWork.bottom = frame->bottom - 2;
130:
131:     loc.p.x = rcWork.left;
132:     GMForeColor(G_WHITE);
133:     loc.p.y = rcWork.bottom;         GMMove(loc);
134:     loc.p.y = rcWork.top;            GMLine(loc);
135:     loc.p.x = rcWork.right;          GMLine(loc);
136:     GMForeColor(G_BLACK);
137:     loc.p.y = rcWork.bottom;         GMLine(loc);
138:     loc.p.x = rcWork.left;           GMLine(loc);
139:
140: // loc.p.y = rcWork.top + 17;        GMMove(loc);  追加
141: // loc.p.x = rcWork.right;           GMLine(loc);
142: // GMForeColor(G_WHITE);
143: // loc.p.y += 2;                     GMMove(loc);
144: // loc.p.x = rcWork.left;            GMLine(loc);
145: }
146:
147: /*
148: ** メニュー項目を描く(全部)
149: */
150: int
151: DrawItems(menu **mhdl, rect *rcWork, point_t pt)
152: {
153:     int         i, cnt, len, ret = 0;
154:     UINT    enableFlag;
155:     char    *ptr;
156:     rect    rWork = {rcWork->left + 3, 0,
157:                             rcWork->right - 2, rcWork->top  + 4};
158: //                          rcWork->right - 2, rcWork->top  + 4 + TITLE_HE
IGHT};
159:     char    enable; /* Bit 0: 許可/不許可
160:                                      Bit 1: 選択/非選択 */
161: // point_t loc;
162:
163:     enableFlag = (**mhdl).mEnable;
164:     cnt  = (**mhdl).mData.mILSize;
165:     ptr  = &(**mhdl).mData.mILDate.mSKey;
166:
167: // GMForeColor(G_BLACK);                           追加
168: // GMBackColor(G_LGRAY);
169: // loc.p.x = rcWork->left + 10;
170: // loc.p.y = rcWork->top +  4;
171: // GMMove(loc);
172: // GMDrawStrL((LASCII *)(*mhdl)->mHandle);
173:
174:     for(i = 1; i <= cnt - 1; i++) {
175:             enableFlag = (enableFlag >> 1);
176:             rWork.top = rWork.bottom;
177:             rWork.bottom += 13;
178:             if((enableFlag & 1) == 0) {
179:                     enable = 0;
180:             } else if(GMPtInRect(&rWork, pt)) {     /* 選択状態 */
181:                     enable = 3;
182:                     ret = i;
183:             } else {                                /* 非選択状態 */
184:                     enable = 2;
185:             }
186:
187:             DrawItem(ptr, i, rcWork, enable);
188:
189:             len = (ptr[2] & (~1));
190:             ptr = &ptr[4 + len];
191:     }
192:
193:     /* 枠を点滅させる */
194:     GMInvertRect(rcWork, 0x888888 >> (counter++ & 7));
195:
196:     return(ret);
197: }
198:
199: /*
200: ** メニュー項目を描く(1項目)
201: */
202: void
203: DrawItem(char ptr[], short no, rect *rcWork, char enable)
204: {
205:     char    title[100];
206:     short   i, len;
207: // point_t loc;
208:
209:     len = (rcWork->right - rcWork->left) / 6;
210:
211:     /* チェックマークを表示 */
212:     if(ptr[1])      *(short *)&title = 0xf603;      /* チェックマーク */
213:     else            *(short *)&title = 0x2020;      /* 空白 */
```

▶朝，仕事(新聞配達)の帰りに山積みになったOh!X(MZ)を発見(1987年6月号より全部)。すべて持ち帰り，楽しませていただいております。早起きは一文(?)の得。
前田　健(19)東京都

```
214:    /* 項目名を表示 */
215:    for(i = 0; i < ptr[2]; i++) {
216:            title[2 + i] = ptr[3 + i];
217:    }
218:    /* 空白を埋める */
219:    for(i = ptr[2] + 2; i < len - 5; i++) {
220: // for(i = ptr[2] + 2; i < len - 4; i++) {
221:            title[i] = ' ';
222:    }
223:    /* ショートカットを表示 */
224:    if(ptr[0]) {
225:            title[i++] = 0xf6;
226:            title[i++] = 0x02;
227:            title[i++] = ptr[0];
228:    } else {
229:            title[i++] = ' ';
230:            title[i++] = ' ';
231:            title[i++] = ' ';
232:    }
233:    title[i++] = ' ';
234:    title[i] = '¥0';
235:    if(!(enable & 2)) {                     /* 非イネーブル */
236:            GMForeColor(G_WHITE);
237:            GMBackColor(G_LGRAY);
238:    } else if(enable & 1) {                 /* 選択状態 */
239:            GMForeColor(G_WHITE);
240:            GMBackColor(G_DGRAY);
241:    } else {                                /* 非選択状態 */
242:            GMForeColor(G_BLACK);
243:            GMBackColor(G_LGRAY);
244:    }
245:    if((enable & 2) && ptr[2] == 0) {
246:            /* へこみを表示 */
247:            rect    rWork = {
248:                    rcWork->left  + 6, rcWork->top + (no-1)* 13 + 6,
249: //                 rcWork->left  + 6, rcWork->top + (no-1)* 13 + 6
250: //                                                 + TITLE_HEIGHT,
251:                    rcWork->right - 6, rcWork->top + (no-1)* 13 + 16
252: //                 rcWork->right - 6, rcWork->top + (no-1)* 13 + 16
253: //                                                 + TITLE_HEIGHT
254:            };
255:            GMBackColor(G_LGRAY);
256:            GMShadowRect(&rWork);
257:    } else {
258:            /* 通常表示             */
259:            point_t loc;
260:
261:            loc.p.x = rcWork->left + 3;
262:            loc.p.y = rcWork->top  + (no-1) * 13 + 4;
263: //         loc.p.y = rcWork->top  + 4 + (no-1) * 13 + TITLE_HEIGHT;
264:            GMMove(loc);
265:            GMDrawStrZ(title);
266:    }
267:
268: }
269:
270: void
271: DoSelect(menu **mhdl, rect *rcWork, short *item)
272: {
273:    short   oldpage;
274:    short   oldfontmode;
275:    short   oldforecolor, oldbackcolor;
276:
277:    int     time, ret;
278:    point_t         pt;
279:
280:    /* グラフィック情報の保存 */
281:    oldpage = GMAPage(3);
282:    oldfontmode         = GMFontMode(G_PSET);
283:    oldforecolor   = GMForeColor(G_LGRAY);
284:    oldbackcolor   = GMBackColor(G_LGRAY);
285:
286:.   time = EMSysTime();
287:    while(EMRBttn()) {
288:            if(EMSysTime() - time > EMDClickGet()) {
289:                    ret = NormalMenu(mhdl, rcWork);
290:                    goto loopout;
291:            }
292:            pt.x_y = EMMSLoc();
293:            DrawItems(mhdl, rcWork, pt);
294:    }
295:    ret = PopupMenu(mhdl, rcWork);
296: loopout:
297:    *item = ret;
298:    if(ret != 0)    revCnt = 8;
299:
300:    /* グラフィック情報を元に戻す */
301:    GMAPage(oldpage);
302:    GMFontMode(oldfontmode);
303:    GMForeColor(oldforecolor);
304:    GMBackColor(oldbackcolor);
305: }
306:
307: int
308: PopupMenu(menu **mhdl, rect *rcWork)
309: {
310:    int     ret = 0;
311:    point_t pt;
312:    event   eventrec;
313:
314:    while(1) {
315:            pt.x_y = EMMSLoc();
316:            ret = DrawItems(mhdl, rcWork, pt);
317:            if(EMGet(EM_MSRDOWN, &eventrec)) {
318:                    /* 右ボタンが押されたか? */
319:                    while(EMRBttn()); /* 右ボタンが放されるまで待つ */
320:                    return(ret);
321:            }
322:            if(EMGet(EM_MSLDOWN, &eventrec)) {
323:                    /* 左ボタンが押されたか? */
324:                    while(EMLBttn()); /* 左ボタンが放されるまで待つ */
325:                            return(ret);
326:            }
327:    }
328: }
329:
330: int
331: NormalMenu(menu **mhdl, rect *rcWork)
332: {
333:    int     ret = 0;
334:    point_t pt;
335:
336:    while(EMRStill()) {
337:            pt.x_y = EMMSLoc();
338:            ret = DrawItems(mhdl, rcWork, pt);
339:    }
340:    return(ret);
341: }
342:
343: /*
344: ** メニュー項目を反転させる
345: */
346: void
347: ReverseItem(menu **mhdl, rect *rcWork, short *item)
348: {
349:    short   oldpage;
350:    short   oldfontmode;
351:    short   oldforecolor, oldbackcolor;
352:
353:    char    *ptr;
354:    int             i, len;
355:
356:    if(*item == 0) return;
357:
358:    /* グラフィック情報の保存 */
359:    oldpage = GMAPage(3);
360:    oldfontmode         = GMFontMode(G_PSET);
361:    oldforecolor   = GMForeColor(G_LGRAY);
362:    oldbackcolor   = GMBackColor(G_LGRAY);
363:
364:    ptr  = &(**mhdl).mData.mILDate.mSKey;
365:    for(i = 1; i < *item; i++) {
366:            len = (ptr[2] & (~1));
367:            ptr = &ptr[4 + len];
368:    }
369:    /* 点滅 */
370:    DrawItem(ptr, *item, rcWork, (revCnt & 1) | 2);
371:
372:    /* グラフィック情報を元に戻す */
373:    GMAPage(oldpage);
374:    GMFontMode(oldfontmode);
375:    GMForeColor(oldforecolor);
376:    GMBackColor(oldbackcolor);
377: }
378:
379: void
380: DoCalc(menu **mhdl)
381: {
382:    int             i, cnt, len, max;
383:    char    *ptr;
384:
385:    /* 初期化処理 */
386:    revCnt = 0;
387:    counter = 0;
388:
389:    /* メニューの幅を求める */
390:    cnt  = (**mhdl).mData.mILSize;
391:    ptr  = &(**mhdl).mData.mILDate.mSKey;
392:
393:    max =0;                                         /* タイトルの文字数 */
394: // max = *(char *)(*mhdl)->mHandle;                /* タイトルの文字数 */
395:    for(i = 1; i <= cnt + 1; i++) {
396: // for(i = 1; i <= cnt - 1; i++) {
397:            len = (ptr[2] & (~1));
398:            if(max < len)   max = len;
399:            ptr = &ptr[4 + len];
400:    }
401:
402:    (*mhdl)->mWidth = (max + 8) * 6 ;
403: // (*mhdl)->mWidth = (max + 8) * 6 + 5;
404:    (*mhdl)->mHight = (cnt - 1) * 13 + 6;
405: // (*mhdl)->mHight = (cnt - 1) * 13 + 6 + TITLE_HEIGHT;
406: }
```

リスト3 MDEF0003.IND

```
1: system.lb
2: -tMDEF -i3 mdef0003.r
```

リスト4 built.BAT

```
1: gcc -O -fomit-frame-pointer -fstrength-reduce mdef0003.c -Wall -c
2: lk -l mdef0003 sxlib.l clib.l gnulib.l doslib.l floatfnc.l
3: ccv mdef0003.x mdef0003.r
4: rlk -l mdef0003.ind system.lb
```

リスト5 MDEF0000.IND

```
1: system.lb
2: -tMDEF -i0 mdef0003.r
```

リスト6 MDEF0001.IND

```
1: a:¥sx¥system.lb
2: -tMDEF -i1 mdef0001.r
```

▶「もうX68000も限界かな」そう思っていました。しかし，5月号の特集記事の数々に新たなX68000の可能性が見えて，久々に目の覚める思いです。EX-Systemが楽しみで，夜も眠れません。いやぁ，Oh!Xって本当に奥が深い！　八木沢　良二(21)栃木県

Oh!X
LIVE in '95

Z-MUSIC ver.2.0
+PCM8用

クリティカルポイント

Utsumi Junichi 内海 淳一

Z-MUSIC ver.2.0
(SC-55対応)

THE SUMMER OF '68

Matsuo Naoki 松尾 直樹

Z-MUSIC ver.2.0
(SC-88対応)

トゥインクルトゥインクル

Hasunuma Masaru 蓮沼 勝

今月は内蔵音源でゲームミュージックっぽいオリジナル曲, SC-55でT-SQUAREのフュージョン, さらにSC-88用に東京競馬トゥインクルレースのイメージソングにもなったWinkの曲をお届けします。SC-88ユーザーの方なら全部再生できます。

AN ADVENTUROUS JOURNEY

まず1曲目は内蔵音源対応のオリジナル曲です。

RPGならば洞窟やダンジョンの暗闇の中を突き進んでいくシーン, シューティングならアステロイド地帯や暗黒星雲のシーン, そんな緊張感のある場面にハマりそうな印象を受けます。とはいってもオドロオドロしいものではなくて, スラップベースとリズムが息をつかせぬようにからみあう実にかっこのいい, スピード感あふれる曲調です。不協和音を巧みに使った進行は, 聞く者をアッと驚かせてくれることでしょう。

また「ダークスペース」('94年11月号掲載)「エスプレッソ銀河」('95年4月号掲載)などのスペーシーなオリジナル曲を聞かせてくれた矢部雅敏君, 彼と似たセンスを持ち合わせているという印象を受けました。試しに「ダークスペース」を聞いたあと, 今回の「クリティカルポイント」を聞いてみてください。MAJORとMINORで調性が反対であるにもかかわらず, なにか共通のテーマが感じられます。

さて, 演奏は内蔵音源対応ですが, ADPCMを多チャンネル再生しているので, 例によってPCM8.Xが必要になります。リスト1のADPCM定義ファイルを入力したら,

A>ZPCNV filename.CNF

でZPDを作成し, 続いてリスト2の演奏データ本体を入力してください。あとは,

A>ZP filename.ZMS

で演奏が始まります。

68の夏?

2曲目はT-SQUAREのアルバム「NEW-S」より「THE SUMMER OF '68」です。THE SUMMER OF '68...なに「68!」と反応してしまいそうですが,感性を光らせる某パソコンとは微塵の関係もないようです。

さてアルバム「NEW-S」というとMEGALITHやLITTLE LEAGUE STARなど元気な曲ばかりが目だっていました。その中でTHE SUMMER OF '68は唯一, アコースティックなギターが泣きの旋律を奏でる実に美しく静かな曲です。原曲では動物の鳴き声や木々が風になびく音などの環境効果音とミックスされていましたが, 今回のデータはさすがにそこまでは再現されていません。しかし,ギターの手弾き感覚はまさに原曲そのもので, コピー曲としては限りなく100%に近い完成度です。

データ制作者の松尾さんは本誌1994年8月号でカシオペアのコピー曲「PURE GREEN」を発表されています。この完成度に驚かれた方はこれを機会にそちらも入力してみましょう。

演奏にはSC-55(GS音源)が必要です。編集部でSC-55mkIIおよびSC-88で正常な演奏を確認しました。

SC-88用データ第1号

3曲目はWinkの「トゥインクルトゥインクル」です。Oh!X LIVEコーナーでは初のSC-88専用曲データです。とはいってもZ-MUSIC ver.2.0ではMIDI同時32チャンネルは制御できないという制限からこの演奏データは単純に音色数と同時発音数の増えたSC-55として使っています。しかしSC-55系にはない音色や, バリエーションがかなり増えた音色(スネアやオーケストラヒットなど)をふんだんに使っているため, ユーザーはかなり自己満足に浸れる(!?)演奏データでもあります。

さて, そのまま演奏するとボーカル(メロディ)パートはピアノ系の音で演奏され, これはこれで楽しいのですが, せっかくの歌謡曲ですからカラオケデータとしても使いたいですよね。そうするにはトラック14(MIDIチャンネル16)をマスクします。具体的にはトラック14をリストから削除(あるいは注釈化)するだけです。

OVERTURE (Wink)

このリストは大変すっきりとしていたのでこのように簡単にカラオケデータ化することができました。歌謡曲をコピーする場合はこのリストのようにボーカルパートを簡単にマスクできる構造にしておくと流用性が高くなっていいと思います。

あと，リスト冒頭でSC-88のモード切り換えを行っています。これはたとえSC-88のパネルでSC-55MAPが選択されていても強制的にSC-88モードにできるメッセージです。これからSC-88用データを作成する人はぜひ参考にしましょう。　(Z.N)

リスト1　クリティカルポイント

```
================  CPOINT.ZMS  ==================
  1: .comment 「Critical point」              Composed by UTSU  199
3/ 8/20
  2:
  3: / 1993/ 8/20, 1994/ 9/22 for ZMUSIC.X + PCM8.X
  4: / INTERNAL
  5:
  6: /----- ADPCM DATA SET ----------------------------------------
  7:
  8: .adpcm_block_data=CPOINT.ZPD
  9:
 10: /----- TRACK SETUP ------------------------------------------
 11:
 12: (i)
 13: (o144)
 14:
 15: (m01,2000)(aFm1,1)
 16: (m02,2000)(aFm2,2)
 17: (m03,2000)(aFm3,3)
 18: (m04,2000)(aFm4,4)
 19: (m05,2000)(aFm5,5)
 20: (m06,2000)(aFm6,6)
 21: (m07,2000)(aFm7,7)
 22: (m08,2000)(aFm8,8)
 23: (m09,2000)(aAdpcm,09)
 24: (m10,2000)(aAdpcm,10)
 25: (m11,2000)(aAdpcm,11)
 26: (m12,2000)(aAdpcm,12)
 27:
 28: /----- OPM DATA SET -----------------------------------------
 29:
 30: /  AR 1DR 2DR  RR 1DL  TL  RS MUL DT1 DT2 AME       Synth1
 31: (@30,       31,  0,  0,  0,  0, 24,  0,  4,  7,  0,  0
 32:     18,  7,  3,  6,  3,  2,  0,  4,  5,  0,  0
 33:     31,  0,  0,  0,  0, 17,  0,  2,  3,  0,  0
 34:     31, 11,  2,  6,  3,  0,  0,  2,  1,  0,  0
 35: /  AL  FB
 36:          4,  7)
 37:
 38: /  AR 1DR 2DR  RR 1DL  TL  RS MUL DT1 DT2 AME       Synth2
 39: (@31,       31,  7,  2,  8,  2, 22,  0,  2,  2,  0,  0
 40:     31,  6,  2,  8,  2, 26,  0,  8,  5,  0,  0
 41:     31,  8,  2,  8,  2, 30,  0,  2,  0,  0,  0
 42:     31, 10,  1,  8,  2,  0,  0,  4,  1,  0,  0
 43: /  AL  FB
 44:     2,  7)
 45:
 46: /  AR 1DR 2DR  RR 1DL  TL  RS MUL DT1 DT2 AME       Synth3
 47: (@32,       31, 11,  4,  6,  3, 28,  0,  4,  7,  0,  0
 48:     18,  9,  5,  8,  3,  2,  0,  4,  7,  0,  0
 49:     31, 13,  3,  6,  3, 22,  0,  2,  3,  0,  0
 50:     31, 13,  2,  8,  3,  0,  0,  2,  3,  0,  0
 51: /  AL  FB
 52:          4,  7)
 53:
 54: /  AR 1DR 2DR  RR 1DL  TL  RS MUL DT1 DT2 AME       Synth4
 55: (@33,       21,  4,  2,  5,  6, 37,  0,  8,  7,  0,  0
 56:     31,  9,  3,  6,  5,  1,  0,  4,  3,  0,  0
 57:     18,  4,  2,  5,  6, 32,  0,  6,  3,  0,  0
 58:     28, 11,  2,  6,  3,  4,  0,  4,  7,  0,  0
 59: /  AL  FB
 60:     4,  7)
 61:
 62: /  AR 1DR 2DR  RR 1DL  TL  RS MUL DT1 DT2 AME       Synth5
 63: (@34,       22,  3,  2,  3,  4, 24,  0,  4,  6,  0,  0
 64:     18,  4,  3,  6,  3,  2,  0,  4,  2,  0,  0
 65:     22,  3,  2,  3,  4, 16,  0,  2,  2,  0,  0
 66:     31, 12,  6,  6,  2,  5,  0,  2,  6,  0,  0
 67: /  AL  FB
 68:     4,  7)
 69:
 70: /  AR 1DR 2DR  RR 1DL  TL  RS MUL DT1 DT2 AME       Bass
 71: (@35,       31, 13,  7,  6,  4, 40,  0,  6,  2,  0,  0
 72:     31, 11,  2,  8,  4, 34,  0,  0,  0,  0,  0
 73:     31,  9,  3,  6,  4, 20,  0,  0,  1,  0,  0
 74:     31, 11,  2,  7,  3,  0,  0,  0,  5,  0,  0
 75: /  AL  FB
 76:     1,  7)
 77:
 78: /  AR 1DR 2DR  RR 1DL  TL  RS MUL DT1 DT2 AME       PSG
 79: (@36,       31,  0,  0,  0,  0, 26,  0,  4,  0,  0,  0
 80:     31,  0,  0,  0,  0, 32,  0,  8,  0,  0,  0
 81:     31,  0,  0,  0,  0, 24,  0,  4,  0,  0,  0
 82:     31, 14,  7,  7,  2,  2,  0,  2,  0,  0,  0
 83: /  AL  FB
 84:     2,  7)
 85:
 86: /  AR 1DR 2DR  RR 1DL  TL  RS MUL DT1 DT2 AME       Synth6
 87: (@37,       18,  0,  2,  6,  0, 27,  0,  4,  7,  0,  0
 88:     24,  0,  3,  7,  0,  7,  0,  6,  5,  0,  0
 89:     21,  0,  1,  4,  0, 15,  0,  2,  3,  0,  0
 90:     27,  0,  2,  6,  0,  4,  0,  8,  1,  0,  0
 91: /  AL  FB
 92:     4,  5)
 93:
 94: /----- MML DATA SET -----------------------------------------
 95:
 96: (t1)        L8v14@k0 @35o2 r2r
 97:     |:4q7er4.q4{ee}q7er4q6er4q7e.q4e.r4:|
 98:     |:|:r4<q7a>q4a<q7g+>q4g+~4q7(d+4.g)_
 99:         r <q7g>q4g<q7f+>q4f+~ q7(d+4.f)_
100:     reree<e>rer|rer4ee:|ereer<e>
101:     ||:4er16er16er16er2er16er16err16<e>:|:|
102:     |:4q7er4.q4{ee}q7er4q6er4q7e.q4e.r4:|
103:     |:4q7erq6c+eq4{ee}q7erq6ee>b<d+q7e.q4e.r<q7e>:|
104:     |:|:r4<q7a>q4a<q7g+>q4g+~4q7(d+4.g)_
105:         r <q7g>q4g<q7f+>q4f+~ q7(d+4.f)_
106:     reree<e>rer|rer4ee:|ereer<e>
107:     ||:4er16er16er16er2er16er16err16<e>:|:|
108:     |:|:4er16er16er16er2er16er16err16<e>:|:|
109:     |:r4<q7a>q4a<q7g+>q4g+~4q7(d+4.g)_
110:       r <q7g>q4g<q7f+>q4f+~ q7(d+4.f)_
111:     reree<e>rer|rer4ee:|ereer<e>
112:     |:|:4q7erq6c+eq4{ee}q7erq6ee>b<d+q7e.q4e.r<q7e>:|:|
113:     e1&e1&e1
114:
115: (t2)        q8v14L4@k3@m8@s6@h44 @30p3o3 r2r8
116:     b1a2<d2>f+1g1b1a2<d2f+1&f+1> |:@31p3
117:     |:a+2.&(a+g+)g2.&(gf)e1&e1:| @32p3
118:     ||:a+g+d+a+8g+d+{a+g+d+g+d+}*120
119:     agda8gd{agd<d>d+}*120:|:| @30p3
120:     b1a2<d2>f+1g1b1a2<d2f+1&f+1> @33p3
121:     |:r{a_8a~}g2{a_a~}g2.r{a_a~}g.r8|g1:|g+1 |:@31p3
122:     |:a+2.&(a+g+)g2.&(gf)e1&e1:| @32p3
123:     ||:a+g+d+a+8g+d+{a+g+d+g+d+}*120
124:     agda8gd{agd<d>d+}*120:|:|
125:     |:|:a+g+d+a+8g+d+{a+g+d+g+d+}*120
126:     agda8gd{agd<d>d+}*120:|:| @31p3
127:     |:a+2.&(a+g+)g2.&(gf)e1&e2.&(eb):| @33p3
128:     |:|:r{a_8a~}g2{a_a~}g2.r{a_a~}g.r8|g1:||a1:|g+1&g+1
129:
130: (t3)        q8v12L4@k-1@m12@s6@h44 @30p1o3 r*32 r2r8
131:     b1a2<d2>f+1g1b1a2<d2f+1&f+1> |:@31p1
132:     |:a+2.&(a+g+)g2.&(gf)e1&e1:| @32p1
133:     ||:a+g+d+a+8g+d+{a+g+d+g+d+}*120
134:     agda8gd{agd<d>d+}*120:|:| @30p1
135:     b1a2<d2>f+1g1b1a2<d2f+1&f+1> @33p1
136:     |:r{a_8a~}g2{a_a~}g2.r{a_a~}g.r8|g1:|g+1 |:@31p1
137:     |:a+2.&(a+g+)g2.&(gf)e1&e1:| @32p1
138:     ||:a+g+d+a+8g+d+{a+g+d+g+d+}*120
139:     agda8gd{agd<d>d+}*120:|:|
140:     |:|:a+g+d+a+8g+d+{a+g+d+g+d+}*120
141:     agda8gd{agd<d>d+}*120:|:| @31p1
142:     |:a+2.&(a+g+)g2.&(gf)e1&e2.&(eb):| @33p1
143:     |:r{a_8a~}g2{a_a~}g2.r{a_a~}g.r8|g1:|a2.&a*20 v13p3
144:     |:r<{c_8c~}>b2<{c_c~}>b2.r<{c_c~}>b.r8|b1:|<c1&c1>
145:
146: (t4)        q8v13L4@k1@m8@s8@h22 @30p3o3 r*4 r2r8
147:     g1f+2a2d+1e1g1f+2a2<d+1>a+1 @31p3
148:     |:f+2.&(f+e)d+2.&(d+c+)c1&c1:| @30p3
149:     |:<d+1g+1a*288&(a2d+)>:| @31
150:     |:f+2.&(f+e)d+2.&(d+c+)c1&c2.&(cg):| @30p3
151:     g1f+2a2d+1e1g1f+2a2<d+1>a+1 @34p3
152:     |:g1&g1|a1&a1:|a1a+2.&(a+f) |:@31p3
153:     |:f+2.&(f+e)d+2.&(d+c+)c1&c1:| @30p3
154:     ||:<d+1g+1a*288&(a2d+)>:|:|
155:     |:|:<d+1g+1a*288&(a2d+)>:|:| @31p3
156:     |:f+2.&(f+e)d+2.&(d+c+)c1&c2.&(cg):| @34p3
157:     |:|:g1&g1|a1&a1:||a*336&(ag):|a1a+2.&(a+f)&f1
158:
159: (t5)        q8v11L4@k-3@m12@s8@h44 @30p2o3 r*40 r2r8
160:     g1f+2a2d+1e1g1f+2a2<d+1>a+1 @31p2
161:     |:f+2.&(f+e)d+2.&(d+c+)c1&c1:| @30p2
162:     |:<d+1g+1a*288&(a2d+)>:| @31p2
163:     |:f+2.&(f+e)d+2.&(d+c+)c1&c2.&(cg):| @30p2
164:     g1f+2a2d+1e1g1f+2a2<d+1>a+1 @34p2
165:     |:g1&g1|a1&a1:|a1a+2.&(a+f) |:@31p2
166:     |:f+2.&(f+e)d+2.&(d+c+)c1&c1:| @30p2
167:     ||:<d+1g+1a*288&(a2d+)>:|:|
168:     |:|:<d+1g+1a*288&(a2d+)>:|:| @31p2
169:     |:f+2.&(f+e)d+2.&(d+c+)c1&c2.&(cg):| @34p2
170:     |:|:g1&g1|a1&a1:||a*336&(ag):|a1a+2.&(a+f)&f1
171:
172: (t6)        q4L16v12@k4@m8@s2 @36p2o4 <a>a<aa>a<aa>a<a>
173:     |:56<a>a<a>a<aa>a<a>a<aa>a<a>a<a>aa<aa>a<aa>a<a>a<aaa>a<
aa>a:|
174:     |:_2<a>a_<a>a_<aa_>a<a_>a<a_a>a_<a_>a_<a>a_a<a_a>a_<aa_>
a<a_>a<a_aa_>a<a_a>a:|
175:
176: (t7)        q4L16v12@k-4@m8@s2 @36p1o4 a_16a~aa_a~aa_a~a
177:     |:56a_a~a_a~aa_a~a_a~aa_a~a_a~a_aa~aa_a~aa_a~a_a~aaa_a~a
a_a~:|
178:     |:_2a_8a~6a_8a~6aa_10a~8a_10a~8a_2a_8a~6a_8a~6a_8a
179:       _2a~8a_2a_8a~6aa_10a~8a_10a~8a_2aa_10a~8a_2a_8a~8:|
180:
181: (t8)        q8v11L4@k6@m8@s12@h66 @37p3o2 r*8 r2r8
182:     |:e*384d+1|d2.&(dd+):|(d+c)&c2.
183:     |:c2.&(cd)d+2.&(d+c+)|{(c+d+)(c+d+)&}2d+*240&(d+c):|(c+d
+)&d+*336
184:     |:c1d+1|f*288&(f2c):|f*384
185:     |:c2.&(cd)d+2.&(d+c+)|{(c+d+)(c+d+)&}2d+*240&(d+c):|(c+d
+)&d+*336
186:     |:e*384d+1|d2.&(dd+):|(d+c)&c2.
187:     |:e*384|f*384:|f1f+2.&(f+c)
```

```
188:     |:|:c2.&(cd)d+2.&(d+c+)|{(c+d+)(c+d+)&}2d+*240&(d+c):|(c
+d+)&d+*336
189:     ||:c1d+1|f*288&(f2c):|f*384:|
190:     |:|:c1d+1|f*288&(f2c):|f*384:|
191:     |:c2.&(cd)d+2.&(d+c+)|{(c+d+)(c+d+)&}2d+*240&(d+c):|(c+d
+)&d+*336
192:     |:e*384|f*384:|f*336&(fe)
193:     |:e*384|f*384:|f1f+2.&(f+c+)c+1
194:
195: (t9)        L8 v9o3 r4.f4
196:     |:4c4d.cc16cd4c4d.c4&c16d4:|
197:     |:4c4dc4cd4c4d4ccdc:|
198:     |:4c4dc4.dcc4d4c4d4:|
199:     |:4c4dc4cd4c4d4|ccdc:|f4d4
200:     |:4c4d.cc16cd4c4d.c4&c16d4:|
201:     |:4c4d.cc16cdc|c4d.c4&c16dc:|f4d.c4&c16dc
202:     |:|:4c4dc4cd4c4d4ccdc:|
203:     ||:4c4dc4.dcc4d4c4d4:|:|
204:     |:|:4c4dc4.dcc4d4c4d4:|:|
205:     |:4c4dc4cd4c4d4|ccdc:|f4d4
206:     |:4c4d.cc16cdcc4d.c4&c16dc:|
207:     |:4c4d.cc16cdc|c4d.c4&c16dc:|f4d.c4&c16df c1
208:
209: (t10)   L16 v9o2 cd8|:7c:|
210:     |:4|:11c:|d8cccd8|:5c:|d8cd8cc|cc:|d8
211:     |:|:10c:|d8|:16c:|d8d8
212:     |e4|:24c:|d8cc:|e4|:21c:|d8cd8cc
213:     |:4cccd8cd8cd8|:8c:|d8cd8cd8cccd8:|
214:     |:|:10c:|d8|:16c:|d8d8
215:     |e4|:18c:|d8ccccd8cc:|e4|:18c:|d8cd8cd8cc
216:     |:4|:11c:|d8cccd8|:5c:|d8cd8cccc:|
217:     |:4|:11c:|d8cccd8|:5c:|d8cd8|cccc:|cd8c
218:     |:|:|:10c:|d8|:16c:|d8d8
219:     |e4|:24c:|d8cc:|e4|:21c:|d8cd8cc
220:     ||:4cccd8cd8cd8|:8c:|d8cd8cd8cccd8:|:|
221:     |:|:4cccd8cd8cd8|:8c:|d8cd8cd8cccd8:|:|
222:     |:|:10c:|d8|:16c:|d8d8
223:     |e4|:18c:|d8ccccd8cc:|e4|:18c:|d8cd8cd8cc
224:     |:|:4|:11c:|d8cccd8|:5c:|d8cd8|cccc:|cd8c:|e1
225:
226: (t11)   L8 v9o1 r4ec{ff}
227:     |:4rdrdrfddrdedcdd|g:|{ff}
228:     |:rdrdrdr{cc}rdrdrdr{cc}rdrdrdrdrd|rdcedg:|cdded{ff}
229:     |:rdrdrdrdrdrdrdd{cc}rdrdrdrdrdrde|dgg:|dc{ff}
230:     |:rdrdrdr{cc}rdrdrdr{cc}rdrdrdrdrd|rdcedg:|edrdg{ff}
231:     |:4rdrdrfddrdedcdd|g:|{ff}
232:     |:4rdrd{cc}f|ddrdedcddg:|{ee}drdrdcdg{ff}
233:     |:|:rdrdrdr{cc}rdrdrdr{cc}rdrdrdrdrd|rdcedg:|cdded{ff}
234:     ||:rdrdrdrdrdrdrdd{cc}rdrdrdrdrdrde|dgg:|dc{ff}:|
235:     |:|:rdrdrdrdrdrdrdd{cc}rdrdrdrdrdrde|dgg:|dc|f:|{ff}
236:     |:rdrdrdr{cc}rdrdrdr{cc}rdrdrdrdrd|rdcedg:|edrdg{ff}
237:     |:4rdrd{cc}fdd|rdedcddg:|{ee}dedcdgf
238:     |:4rdrd{cc}f|ddrdedcddg:|{ee}drdrdcdg{ff}
239:
240: (t12)   L8 v9o1 r2r
241:     |:4a4rr{aa}a4ra4ra.a.rr:|
242:     |:|:2rraaaaa4rraaaaa4rraraaarar|rarraa:|araara
243:     ||:4a.a.a.ar2a.a.ar.a:|:|
244:     |:4a4rr{aa}a4ra4ra.a.rr:|
245:     |:4a4aa{aa}a4aaaaa.a.ra:|
246:     |:|:2rraaaaa4rraaaaa4rraraaarar|rarraa:|araara
247:     ||:4a.a.a.ar2a.a.ar.a:|:|
248:     |:|:4a.a.a.ar2a.a.ar.a:|:|
249:     |:2rraaaaa4rraaaaa4rraraarrar|rarraa:|araara
250:     |:|:4a4aa{aa}a4aaaaa.a.ra:|:| a1
251:
252: (p)
```

リスト2　クリティカルポイント用コンフィグファイル

```
===============  CPOINT.CNF  ==================
  1: /   Set Rhythm
  2: /   [ CPOINTFM.CNF ]
  3:
  4: 1 = u110bd1.pcm, v110
  5: 2 = elcs.pcm, v70
  6:
  7: .o1c = tamb.pcm, v177
  8: .o1d = shakerm1.pcm, v200
  9: .o1e = hiq.pcm, v77
 10: .o1f = slap2.pcm, v55
 11: .o1g = clp808.pcm, v70
 12: .o1a = hlt1.pcm, v65
 13:
 14: .o2c = CHH78.pcm, v60
 15: .o2d = tr808ho.pcm, v77, m36
 16: .o2e = crash11.pcm, v120
 17:
 18: .o3c = ctrlk.pcm, v70, m1
 19: .o3d = rvbs2.pcm, v90, m2
 20: .o3f = gate_sd.pcm, v60, m2
 21:
 22: .erase 1
 23: .erase 2
```

リスト3　クリティカルポイント用カウンタ表示

```
 1:000056B8 00000000   2:00005538 00000000   3:0000553C 00000000   4:0000553C 00000000
 5:00005560 00000000   6:0000576C 00000000   7:0000576C 00000000   8:00005540 00000000
 9:00005538 00000000  10:00005538 00000000  11:00005478 00000000  12:00005538 00000000
```

リスト4　THE SUMMER OF '68

日本音楽著作権協会(出)許諾第9570183-501号

```
==================  SUMMER_6.ZMS  ==================
  1: .comment ** THE SUMMER OF '68 **  T-SQUARE for SC55  by  ﾏｯ
ﾁｭﾝ
  2: (i)(b1)
  3: (M1,4200)(AMIDI1,1)
  4: (M2,3700)(AMIDI2,2)
  5: (M3,4000)(AMIDI3,3)
  6: (M4,4810)(AMIDI4,4)
  7: (M5,2800)(AMIDI5,5)
  8: (M6,4800)(AMIDI6,6)
  9: (M7,2800)(AMIDI7,7)
 10: (M8,2800)(AMIDI8,8)
 11: (M9,1200)(AMIDI9,9)
 12: (M10,4830)(AMIDI10,10)
 13: (M11,3000)(AMIDI11,11)
 14: (M12,3000)(AMIDI12,12)
 15: (M13,2200)(AMIDI13,13)
 16: (M14,2000)(AMIDI14,14)
 17: (M15,2200)(AMIDI15,15)
 18: (M16,2000)(AMIDI10,16)
 19: (M17,2700)(AMIDI16,17)
 20: (M18,1000)(AMIDI10,18)
 21:
 22: .ROLAND_EXCLUSIVE $10,$42={$40,$00,$7F,$00} /音源初期化
 23: .ROLAND_EXCLUSIVE $10,$42={$40,$11,$30,$38} /Vibrato Rate
 24: .ROLAND_EXCLUSIVE $10,$42={$40,$19,$30,$38} /Vibrato Rate
 25: .ROLAND_EXCLUSIVE $10,$42={$40,$1F,$30,$38} /Vibrato Rate
 26: .ROLAND_EXCLUSIVE $10,$42={$40,$11,$31,$4F} /Vibrato Depth
 27: .ROLAND_EXCLUSIVE $10,$42={$40,$19,$31,$4F} /Vibrato Depth
 28: .ROLAND_EXCLUSIVE $10,$42={$40,$1F,$31,$4F} /Vibrato Depth
 29: .ROLAND_EXCLUSIVE $10,$42={$40,$11,$37,$50} /Vibrato Delay
 30: .ROLAND_EXCLUSIVE $10,$42={$40,$19,$37,$52} /Vibrato Delay
 31: .ROLAND_EXCLUSIVE $10,$42={$40,$1F,$37,$52} /Vibrato Delay
 32: .ROLAND_EXCLUSIVE $10,$42={$40,$12,$34,$42} /TVF&TVA attack
 33: .ROLAND_EXCLUSIVE $10,$42={$40,$13,$34,$42} /TVF&TVA attack
 34: .ROLAND_EXCLUSIVE $10,$42={$40,$14,$34,$42} /TVF&TVA attack
 35: .ROLAND_EXCLUSIVE $10,$42={$40,$16,$34,$41} /TVF&TVA attack
 36: .ROLAND_EXCLUSIVE $10,$42={$40,$1B,$34,$43} /TVF&TVA attack
 37: .ROLAND_EXCLUSIVE $10,$42={$40,$1C,$34,$41} /TVF&TVA attack
 38: .ROLAND_EXCLUSIVE $10,$42={$40,$12,$35,$48} /TVF&TVA Decay
 39: .ROLAND_EXCLUSIVE $10,$42={$40,$12,$32,$38} /TVFcutoff freq
 40: .ROLAND_EXCLUSIVE $10,$42={$40,$16,$32,$3B} /TVFcutoff freq
 41: .ROLAND_EXCLUSIVE $10,$42={$40,$12,$33,$50} /TVF resonance
 42: .ROLAND_EXCLUSIVE $10,$42={$40,$16,$33,$3F} /TVF resonance
 43: .ROLAND_EXCLUSIVE $10,$42={$40,$11,$36,$42} /TVF&TVArelease
 44: .ROLAND_EXCLUSIVE $10,$42={$40,$1A,$15,$01} /Use Drumpart
 45: .ROLAND_EXCLUSIVE $10,$42={$40,$1D,$15,$01} /Use Drumpart
 46:
 47:
 48: .sc55_v_reserve={2,2,4,4,1, 1,1,2,2,3, 2,0,0,0,0, 0}
 49:
 50: .SC55_reverb $10={4,2,2,118,102,35,00}     / Reverb set
 51: .SC55_chorus $10={2,0,68,10,68,03,22,00}   / Chorus set
 52:
 53:
 54: (t1)t76o4@v125q8@p48@g12@u120L16@25k-12
 55:                   m2s0h0@h24@s5@j1         / Guitar
 56: (t2)o4q8@v100@p63@g12@u100L16@97           / Bell
 57: (t3)o4@q4@v114@p42@g12@u70L16@51@c11,127,127
 58:                   m,1s,2h,0@h,0@s,89@a-64 / Strings
 59: (t4)o4@q4@v127@p82@g12@u70L16@51@c11,127,127
 60:                   m,1s,2h,0@h,0@s,89@a-64 / Low Strings
 61: (t5)o2q8@v102@p42@g12@u100L16@5            / E.piano & Pf
 62: (t6)o2@q2@v122@p64@g12@u110L16@34          / Bass
 63: (t7)o5q7@v112@p64@g12@u110L16@12           / Vib.
 64: (t8)o4q7@v116@p44@g12@u110L16@74           / Flute
 65: (t9)o4q8@v72@p78@g12@u120L16@25k-12
 66:                   m2s0h0@h24@s5@j1         / Gt. delay
 67: (t10)o2@v110@r1@g12@u100L16@33             / drum
 68: (t11)o2@v98@r1@g12@u127L16                 / kick & click
 69: (t12)o4q8@v126@p100@g12@u100L16@118        / Talking Drum
 70: (t13)o7q8@v95@p54@g12@u80L16i127,0@101@c11,127,127
 71:                   m,1s,3h,1@h,24@s,98      / Wind Chime
 72: (t14)o4@v92@r1@g12@u100L16                 / Perc.
 73: (t15)o4q7@v87@p60@g12@u100L16@14           / Wood wind
 74: (t16)o4@r1@u100L16                         / Maracas
 75: (t17)o4@v105q8@p48@g12@u120L16@25k-12
 76:                   m2s0h0@h24@s5@j1         / Guitar II
 77: (t18)o3@r1@u100L16                         / drum Cup
 78:
 79:
 80: .sc55_print "The Summer of'68"
 81:
 82: (t1)@i$41,$10,$42@e79,48
 83: (t2)@i$41,$10,$42@e106,10
 84: (t3)@i$41,$10,$42@e98,42
 85: (t4)@i$41,$10,$42@e48,55
 86: (t5)@i$41,$10,$42@e75,48 y$7f,0 w
 87: (t6)@i$41,$10,$42@e32,20w
 88: (t7)@i$41,$10,$42@e78,40w
```

▶そうか，Oh!Xは世界で唯一，５インチFDが付録につく雑誌だったのか……。
阪長　俊之(28)京都府

```
 89: (t8)@i$41,$10,$42@e60,15w
 90: (t9)@i$41,$10,$42@e106,28w
 91: (t10)@i$41,$10,$42@e118,28
 92:
 93: @y$18,38,$3f @y$18,41,$3e @y$18,55,$3e @y$18,36,$39/ﾁｭｰﾆﾝｸﾞ
 94: @y$18,49,$41 @y$18,57,$41 @y$18,55,$3f
 95: @y$1a,46,$3F                                          /TVAﾚﾍﾞﾙ
 96: @y$1d,38,108 @y$1d,37,94  @y$1d,53,78 @y$1d,46,126 /ﾘﾊﾞｰﾌﾞ
 97: @y$1e,52,$6B                                          /ｺｰﾗｽ
 98: @y$1c,71,120 @y$1c,72,115 @y$1c,55,72                 /ﾊﾟﾝﾎﾟｯﾄ
 99:                       w
100:
101: (t11)@i$41,$10,$42@e26,46
102:             @y$1d,81,100 @y$1d,81,118 x$40,$1a,$14,0
103:
104: (t12)@i$41,$10,$42@e25,58   y$7e,1
105: (t13)@i$41,$10,$42@e124,10
106: (t14)@i$41,$10,$42@e48,48 @y$18,60,$3e @y$18,61,$3e/ﾁｭｰﾆﾝｸﾞ
107:                     w
108: (t15)@i$41,$10,$42@e68,44w
109: (t16)                     w
110: (t17)@i$41,$10,$42@e88,32w
111: (t18)                     w
112:
113:
114: /------------------------- ｲﾝﾄﾛ  (A) ------------------------
115: /イントロ前2小節追加
116: (t1)r1
117:
118: (t1)r1r1r1
119: (t1)r4.o5z100,88ef@u112f&g8.&g8z100,88fe
120: (t1)z110,,100,88,90,100,78,88,78,90e*4&f*32e8.c2>a<c
121: (t1)>b4.geg2
122:
123: (t1)r1
124: (t1)r4.o5z100,88ef@u112f&g8.&g8z100,88fe
125: (t1)z110,,100,88,90,100e*4&f*32e8.c2>a<c
126: (t1)z78,,98c*4&d*68b8@u100f&g4..  r1
127:
128: (t1)r4.z100,88,100,,110agg&a8<c^4>
129: (t1)r4.z100,88,100,,108,,100,88agg&a8<d&e8dc
130: (t1)z95,,105,,88c&d^4d&e>a2  r1  r2..z88,98ga
131:
132: (t1)z108,98,88,100,100<c4.>agg&a4..
133: (t1)r2..z88,98,108,98,88,100,100ga  <d4.>agg&a4..  r1r1
134:
135:
136: (t2)r1  z40,60|:12o5{c<c>c<c>c<c}4:|
137: (t2)|:2|:4o5{c<c>c<c>c<c}4:|  |:4o5{dfdfdf}4:|
138: (t2)|:8o4{a<g>a<g>a<g>}4:|:|
139: (t2)|:8o4{a<a>a<a>a<a>}4:|  |:8o5{dgdgdg}4:|
140: (t2)|:8o5{c<c>c<c>c<c}4:|  |:16o4{a<a>a<a>a<a>}4:|
141:
142: (t3)r1  @u88o4'cg'*1344  'cg'*768
143: (t3)'cg'*384  'ea'*384 'g1b'&  'g2b'z98'g2b' @u82'ga<c'*768
144:
145: (t4)r1
146: (t4)r1r1r1  |:2@u98o2'd1^1<f'  'a1^1<g<d':|
147: (t4)'f1^1<a'  @u80'a1^1<fa<d'  'a1^1<g<ce'  'g<a<df'*768
148:
149:
150: (t11)r1  r1r1r*3264  o5@u50a1  @u127o2r1
151:
152:
153: (t12)r1
154: (t12)r1r1r1r1  o2r4.@u100@b0,5145,1g8r2  r1  @p27r4.g8r2
155: (t12)r1r1r1  @p100r4.g8r2  r1
156: (t12)@p27r2r8g8r4  r2.@p100g8r8  r1r1  r4.@p27g8r2
157: (t12)r1  @p100r2r8g8r4  r1r1
158:
159: (t13)r1  r1r1r*3264  o8r4@a127
160: (t13){fedc>bagfedc>bagfedc>bagfedc>bagfedc>bagfedc>bagfedc>
bagf}2.^1
161:
162: /---------------------  A (B)  --------------------------
163:
164: (t1)o5r4.z88,98,100,,98,88efg&g^4fe
165: (t1)z108,,98,88,80,90,78e*4&f*32e8.c2>a<c
166: (t1)>b4.z98,88,100,100geg2&  g1
167: (t1)r4.z88,98,100,,98,88<efg&g^4fe
168:
169: (t1)z108,,98,88,80,90,78e*4&f*32e8.c2>a<c
170: (t1)z98,,104,90,,90c&d^4b8f*4&g*92&  g1
171:
172:
173: (t2)|:2|:4o5{c<c>c<c>c<c}4:|  |:4o5{dfdfdf}4:|
174: (t2)|:8o4{a<g>a<g>a<g>}4:|:|
175:
176: (t3)@u88o4|:2'cg'*768:|
177:
178: (t4)o2z98,94|:2'd1^1<f'  'a1^1<g<d':|
179:
180: (t6)@u100d1  d1  z100,88,100>a2...e  a1
181: (t6)@u100<d1  d1  z100,88,100,110>a2...e  a2.g4
182:
183: (t10)|:2z40,60,80,100,60f+4f+4f+*0&c+4f+4
184: (t10)z80,100,80,60,,80,100,60,,,60
185: (t10)f+8a+8f+8f+f+f+*0&c+4f+f+f+f+  z80,60,80,100,60
186: (t10)f+4f+4f+*0&c+4f+4  z80,100,80,60,,80,100,60,,,60
187: (t10)f+8a+8f+8f+f+f+*0&c+4f+f+f+f+:|
188:
189: (t11)w6w10w14z127,50,127c4<<<a2.>>>  c1
190: (t11)z50,127,127|:3o5a*0&o2c1  c1:|
191:
192: (t12)|:4@p27r2rb8rg8r8  @p100r2.g8r8:|
193:
194: (t13)r1r1r1r1r1r1r1r1
195:
196: (t14)z78,,98o5|:4r2..a+a+  r2.d+4:|
197:
198: /---------------------  B (B)  --------------------------
199:
200: (t1)r4.z100,88,100,,110agg&a<c4.>
201: (t1)r4.z100,88,100,,108,,98,88agg*4&a*32<d&e8dc
202:
203: (t1)z95,,105,,88c&d^4d&e>a2  r1  r2..z88,98ga
204: (t1)z108,98,88,100,,100<c4.>agg&a4..&
205: (t1)a2.r8z88,98,108,98,88,100,100ga  <d4.>agg&a4..
206:
207: (t2)|:8o4{a<a>a<a>a<a>}4:|  o5|:8{dgdgdg}4:|
208: (t2)|:8{c<c>c<c>c<c>}4:| |:8o4{a<a>a<a>a<a>}4:|w
209:
210: (t3)@u88o4'cg'*384
211: (t3)'ea'*384  'g1b'&'g2b'z98'g2b'  @u82'ga<c'*384
212:
213: (t4)o2@u98'f1^1<a' @u80'a1^1<fa<d' 'a1^1<g<ce' 'g<a<df'*384
214:
215: (t6)@u100o1f1  f1  z100,90,100g2...d  g1  a2...e  a1 g2...d
216: (t6)g2^8.z112,102,92<gd8g8
217:
218: (t10)|:3z80,60,80,100,60f+4f+4f+*0&c+4f+4
219: (t10)z80,100,80,60,,80,100,60,,,60
220: (t10)f+8a+8f+8f+f+f+*0&c+4f+f+f+f+:|
221: (t10)z80,60,80,100,60f+4f+4f+*0&c+4f+4
222: (t10)z80,100,80,60,,40,100,120,110f+8a+8f+8f+f+f+*0&c+8.a8.
d8
223:
224: (t11)z50,127,127|:4o5a*0&o2c1  c1:|
225:
226:
227: (t12)|:4@p27r2rb8rg8r8  @p100r2.g8r8:|w
228:
229: (t13)r1r1r1r1r1r1r1
230: (t13)o8@a@s70@a127{fedc>bagfedc>bagfedc>bagfedc>bagfedc>bag
f}1
231:
232: (t14)z78,,98|:4r2..a+a+  r2.d+4:|
233:
234: /-----------------------  C  ---------------------------
235:
236: (t1)o5z100,90,80,85,100,90,82'g8.<e-''f8.<d''d8b-''e-8.<c'
237: (t1)'g8.<e-''f8<d'  'e-2.<c'z90,85{r'db-''e-<c'}4
238: (t1)z100,88,80'f8.<d'g8.'d8^2b-'
239:
240: (t1)r2.z108,98,,100,110,95{rb-g&}4  f2.{re-f}4  c2.{rb-g&}4
241: (t1)f2.{re-f}4  g2z100,,95,88f*4&g*32g8.g8
242:
243:
244: (t3)o6z100,,98,104,108,112,110c1&  c4d4e-8f4<c8>  b-1
245: (t3)z100,95,90,85,80,100,90,102>'b-2<g''b-2<d' 'g2b-''f2a-'
246: (t3)'e-2g'f8e-4f8  >@u88'b-1<d'&  'b-2<d'<@u86'f2b-'
247:
248: (t4)@u98o3'a-1^1<c'  'b-1<d'&
249: (t4)'b-1<d'  'a-1^1<c'  @u80'c1^1e-gb-<d'
250:
251: (t5)o4r1r1  r2r8z100,88'b-<dg'rr4  'gb-<df'r2... r1r1r1  r1
252:
253:
254: (t6)o1z100,90,100a-4..a-r8a-4.  b-4..b-r8b-4.
255: (t6)z100,90<e-4..e-r8e-^4>b-  <e-4..e-r8e-4e-8
256: (t6)d-4..d-r8d-^4>a-  z100,90,100<d-4..d-r8d-4.  c4..cr8c4.
257: (t6)z100,90,80,90,100,90,80,95c4..cr>b-<cgc>b-g8
258:
259:
260: (t7)o5z100,90,80,85,100,90,82e-8.d8.>b-8<c8.e-8.d8
261: (t7)c2.z90,85>{b-<cr}4  z100,88,80d8.>g8.b-8^2
262: (t7)r1  z88,108,98o5|:2r8f8.g8.<c2>:|  |:2r8f8.g8.<d2>:|
263:
264: (t8)r1r1r1r1  z60,65,70,75,70,65,60
265: (t8)o4|:2{rrr'd-f''e-g''fa-''gb-''fa-''e-g'}4'fd-8'
266: (t8)z65,60{fd-fd-fd-fd-fd- fd-fd-fd-fd-fd-}2r8 :|
267: (t8)z60,65,70,75,80,75,70,65,60,60
268: (t8)|:2{rrr'df''e-g''fa-''gb-''a-<c''gb-'}4{'fa-''e-g''df'&
}8
269: (t8)'d8f'z65,60{fdfdfdfdf dfdfdfdfd}2:|
270:
271:
272: (t10)z114,60,,80,110,60,,80,60,,100,80,110,60,,60
273: (t10)o3c+8>f+f+f+*0&df+f+f+ f+f+a+8 f+*0&df+f+f+
274: (t10)z80,60,,,80,110,60,,80,60,,100,80,110
275: (t10)|:3f+f+f+f+ f+*0&df+f+f+ f+f+a+8 f+*0&d@u60f+f+f+:|
276:
277: (t10)z80,60,,,80,110,60,,80,60,,100,80,120
278: (t10)|:3f+f+f+f+ f+*0&df+f+f+ f+f+a+8 f+*0&f@u60f+f+f+:|
279: (t10)z80,60,,,80,110,60,,80,60,,80,,120,60,80
280: (t10)f+f+f+f+f+*0&df+f+f+ f+f+f+f+*0&<c>f+f+z80,127,60f+*0&
gf+
281:
282:
283: (t11)w5w7w8w15w16o2@u127|:7c4..c8.c4.:|  c4..c8.c8.c8.
284:
285: (t13)r1r1r1r1 r1r1r1r1
286:
287: (t14)o4z100,90,110|:4r1  r4.c+c+c2:|
288:
289:
290: (t15)z70,75,80,75,70,85
291: (t15)o3r'a-<c''b-<d'<'ce-'r8>'b-<d'r'a-8.<c''b<d'r4
292: (t15)r'a-<c''b-<d'<'ce-'r8>'b-<d'r'a-8.<c'<'ce-'>r4
293: (t15)r1  z70,75,80,85,75,70,85
294: (t15)'b-<d''b-<d'<'ce-''df'r8>'b-<d'r'g8.b-''g<c'r4r1r1r1r1
295:
296: (t16)o4z70,50,70|:32a+8a+a+:|
297:
298: /-----------------------  D  ---------------------------
299:
300: (t1)o5z100,90,80,85,100,90,82
301: (t1)'g8.<e-''f8.<d''d8b-''e-8.<c''g8.<e-''f8<d'
302: (t1)'e-2.<c'z90,85{r'db-''e-<c'}4 z100,88,80'f8.<d'g8.'d8^2
b-'
303:
304: (t1)r2.z108,98,,100,110,95{rb-g&}4  f2.{re-f}4
305: (t1)c2.z114,98,106,106{rb-g}4  b-*4&<c*140>{rb-<c}4
306: (t1)e-2&e-8z110,88g4.  <c1^1
```

▶Xellent30sがX68000PROに対応していないではないか！　買おうと思っていたのに……。
小野寺　健一(18)宮城県

```
307:
308:
309: (t3)o6z100,,98,104,108,112,110c1&  c4d4e-8f4<c8>  b-1
310: (t3)z100,95,90,85,80,100,90,102>'b-2<g''g2<d'  'g2b-''f2a-'
311: (t3)'e-2g'f8e-4f8  @u98g1&  g1&  g1  r1
312:
313: (t4)@u98o2@d1'a-1<a-<c'  @u100b-1@d0  @u88o3'e-1b-1<d'&
314: (t4)'e-b-1<d'  'd-a-1^1<c'  @u80>'a-1^1^1^1<e-a-<c'
315:
316: (t5)o4r1r1  r2r8z100,88'b-<dg'rr4  'gb-<df'r2...w
317:
318:
319: (t6)o1z100,90,100a-4..a-r8a-4.  b-4..b-r8b-4.
320: (t6)z100,90<e-4..e-r8e-^4>b-  <e-4..e-r8e-8.>b-<e-8
321: (t6)z100,90d-4..d-r8d-^4>a-  z100,90,100<d-4..d-r8d-4.
322: (t6)>a-4..a-r8a-4.  a-4..a-r8a-4.^1^1
323:
324:
325: (t7)o5z100,90,80,85,100,90,82e-8.d8.>b-8<c8.e-8.d8
326: (t7)c2.z90,85>{b-<cr}4 z100,88,80d8.>g8.b-8^2 r1 z88,108,98
327: (t7)o5|:3r8f8.g8.<c2>:|  r8b-8.<c8.g2  @u105<c1&  c1w
328:
329: (t8)r1r1r1r1  z60,65,70,75,70,65,60o4
330: (t8)|:2{rrr'd-f''e-g''fa-''gb-''fa-''e-g'}4'fd-8'z65,60
331: (t8){fd-fd-fd-fd-fd- fd-fd-fd-fd-fd-}2r8 :|
332: (t8)z60,65,70,75,80,75,70,65,60,60
333: (t8)|:2{rrr'df''e-g''fa-''gb-''a-<c''gb-'}4{'fa-''e-g''df'&
}8
334: (t8)'d8f'z65,60{fdfdfdfdf dfdfdfdfd}2:|w
335:
336:
337: (t10)z114,60,,80,110,60,,80,60,,100,80,110,60,,60
338: (t10)o3c+8>f+f+f+*0&df+f+f+ f+f+a+8 f+*0&df+f+f+
339: (t10)z80,60,,,80,110,60,,80,60,,100,80,110
340: (t10)|:3f+f+f+f+ f+*0&df+f+f+ f+f+a+8 f+*0&d@u60f+f+f+:|
341:
342: (t10)z114,60,40,80,110,60,,80,60,,108,80,110,60,,60
343: (t10)<a8>f+f+ f+*0&df+f+f+ f+f+a+8 f+*0&df+f+f+
344: (t10)z80,60,,,80,110,60,,80,60,,100,80,120
345: (t10)|:2f+f+f+f+ f+*0&df+f+f+ f+f+a+8 f+*0&f@u60f+f+f+:|
346: (t10)z80,60,,,80,110,60,,80,60,,80,60,,114
347: (t10)f+f+f+f+ f+*0&df+f+f+ f+f+f+8 f+f+<f8
348: (t10)z118a4x$40,$10,$14,2r4
349: (t10){@u10c+c+@u15c+c+@u20c+c+@u25c+c+@u30c+c+@u35c+c+@u40c
+c+@u50c+c+@u55c+c+@u60c+c+@u65c+c+@u70c+c+@u75c+c+@u80c+c+@u85c
+c+@u70c+c+@u60c+}1
350: (t10)r2x$40,$10,$14,0
351:
352: (t11)o2@u127|:8c4..c8.c4.:|  r1r1
353:
354: (t13)r1r1r1r1r1r1
355: (t13)o8@a@a127{fedc>bagfedc>bagfedc>bagfedc>bagfedc>bagf}1w
356:
357: (t14)o4z100,90,110|:4r1  r4.c+c+c2:|  r1r1
358:
359: (t15)z70,75,80,75,70,85
360: (t15)o3r'a-<c''b-<d'<'ce-'r8>'b-<d'r'a-8.<c''b<d'r4
361: (t15)r'a-<c''b-<d'<'ce-'r8>'b-<d'r'a-8.<c'<'ce-'>r4  r1
362: (t15)z70,75,80,85,75,70,85
363: (t15)'b-<d''b-<d'<'ce-''df'r8>'b-<d'r'g8.b-''g<c'r4w
364:
365: (t16)o4z70,50,70|:32a+8a+a+:|w
366:
367: /-----------------------  E  ----------------------------
368: (t9)r2
369: (t1,9)z112,,100,,108,98,88o5r4f&g^4f8&{fed}4
370: (t1,9)>a2<rz112,,102,92,110,,100,90,80c*4&d*8eff*4&g*8fed
371: (t1,9)z100,90,78,88,88c8.>b8.ge&e2
372: (t1,9)r4>z88,90,92,94,96,98,100,102,98,106,110ab<cdefgab&<c
de
373: (t1,9)z100,,88,,92,,95,98>b*4&<c*32>a&a2b&<cde
374: (t1,9)z108,,88,100,95,,98,101,104>b*4&<c*32>a8.<d^4c+&defg
375: (t1,9)z110,,100,88,104,88a+*4&b*32g8ea2e8
376: (t1,9)z104,114,100,78,95,100,88,100,110b8<c>g8ea4.aeb<c
377:
378: (t2)z40,60|:2|:4o5{c<c>c<c>c<c}4:|  |:4o5{dfdfdf}4:|
379: (t2)|:8o4{a<g>a<g>a<g>}4:|:|
380:
381: (t3)@u88o4|:2'cg'*768:|
382:
383: (t4)z98,92o2|:2'd<f'*384  'a<g<d'*384:|
384:
385: (t6)o2z100,90d2...>a  <d2..d8>  a2...e  a1
386: (t6)z100,90,100,90,95<d2...>a  <d2..>gg+  z100,90a2...e
387: (t6)z100,95,100,105,110,100,110a^4<egg+ae>@b0,-8191,48a4.@b
0
388:
389:
390: (t10)z40,60,,80,100,60,60o2f+4f+f+r8f+*0&c+4f+f+r8
391: (t10)z80,100,80,60,40,100,60,,,60f+8a+8f+8f+8f+*0&c+4f+f+f+
f+
392: (t10)z40,60,,80,100,60,60|:2f+4f+f+r8f+*0&c+4f+f+r8
393: (t10)z80,100,80,60,,80,100,60,,,60
394: (t10)f+8a+8f+8f+f+f+*0&c+4f+f+f+f+:|
395: (t10)z40,60,,80,100,60,60f+4f+f+r8f+*0&c+4f+f+r8
396: (t10)z80,100,80,60,,40,100,60,60,40,60,60
397: (t10)f+8a+8f+8f+f+f+*0&c+f+f+8f+f+8f+
398:
399: (t11)w2w9w12z50,127,127|:4o5a*0&o2c1  c1:|
400:
401: (t12)@b0,5145,1@u100|:4@p27r2rb8rg8r8  @p100r2.g8r8:|
402:
403: (t14)z78,,98o5|:4r2..a+a+  r2.d+4:|
404:
405: /---------------------  F (B)  --------------------------
406:
407: (t1,9)z100,,88,100,90,95c&d8>a^2b8<cd
408: (t1,9)z108,,60,95,90,,95,100,105d&e>aa2g+&ab<cd
409: (t1,9)z108,,85,75,70,80,88,,,90,85,90,95,100,,75
410: (t1,9)d&e8>@q0a*6f*6@q2ef{e&f&e}16dc>b<cd>g*4&a*20g@u88g&
411: (t1,9)g^4z75,80,85,90,95,100,105,90,95,100f8gab<cdecde
412: (t1,9)z85,110,100,75,88,108,95,85,90,95,100e8<c>b8ga4<c>eab
<cd
413: (t1,9)z105,,110,98,78,92,78,88,,,80,75,70
414: (t1,9)d&eed8.>b<c4c>@q0{b&<c&>b}16a@q2ge
415: (t1,9)z78fz75,85,95efg8fga8gab8gab
416: (t1,9)z100<c>z90,95,100b<cd8cde8def^4
417:
418:
419: (t2)|:8o4{a<a>a<a>a<a>}4:|  o5|:8{dgdgdg}4:|
420: (t2)|:8{c<c>c<c>c<c>}4:| |:8o4{a<a>a<a>a<a>}4:|w
421:
422: (t3)@u88o4'cg'*384
423: (t3)'ea'*384  'g1b'&'g2b'z98'g2b'  @u82'ga<c'*384
424:
425: (t4)o2@u98'f1^1<a' @u80'a1^1<fa<d' 'a1^1<g<ce' 'g<a<df'*384
426:
427: (t6)o1@u100f1  f1  z100,90,100,90,80,100,90g2...d  g2..gg+
428: (t6)f2...e  z100,,110,100,95,90a1  g2...<d  g2d4d4
429:
430:
431: (t10)z60,80,,100,60,60r8f+8f+4f+*0&c+4f+f+8.
432: (t10)z80,100,80,60,,80,100,60,40,,40
433: (t10)f+8a+8f+8f+f+f+*0&c+4f+f+f+f+  z80,60,,80,100,60,60
434: (t10)f+4f+f+r8f+*0&c+4f+f+r8  z80,100,80,60,,80,100,60,,,,1
00
435: (t10)f+8a+8f+8f+f+f+*0&c+8f+f+f+f+a+8
436: (t10)z80,60,60,80,60,40,100,60,,80f+8f+8f+f+8f+f+*0&c+f+f+8
f+4
437: (t10)z80,60,40,80,60,80,100,40,80,40,40
438: (t10)f+8f+f+8f+8f+f+*0&c+8f+8f+8f+f+
439: (t10)z40,60,80,60,,80,100,60,,80f+8f+8f+f+8f+f+*0&c+f+f+8f+
4
440: (t10)z80,100,80,60,,80,110,80,124,60,120
441: (t10)f+8a+8f+8f+f+f+*0&d8f+<c8>f+g8
442:
443: (t11)w17z50,127,127|:4o5a*0&o2c1  c1:|
444:
445: (t12)|:4@p27r2rb8rg8r8  @p100r2.g8r8:|w
446:
447: (t14)z78,,98o5|:4r2..a+a+  r2.d+4:|
448:
449: (t17)o5z100,90,80,112,,80,100,82,,90,95,100,110,110r2.g8ab
450: (t17)b&<c>ff2e&fgab  b&<c8rr2.  r1r1
451: (t17)o5z110,,100,88,80,92b&<cc>b8.ga4rr4
452: (t17)>z100,90,100,108,92,98,104,92,98,104,92,98,104,110
453: (t17)agab8ab<c8>b<cd8>b<cd  ez95,105,115def8efg8fga^4
454:
455: /-------------------------------------------------------
456: /--------------------  Repeat  -------------------------
457: /-------------------------------------------------------
458:
459: /-----------------------  C  ---------------------------
460:
461: (t1)o5z100,90,80,85,100,90,82'g8.<e-''f8.<d''d8b-''e-8.<c'
462: (t1)'g8.<e-''f8<d'  'e-2.<c'z90,85{r'db-''e-<c'}4
463: (t1)z100,88,80,100'f8.<d'g8.'d4.b-'r8.f
464: (t1)z105,115,105,115,100,88,,80e-fe-fc>b-&<c>b-^4
465: (t1)z108,98,,100,110,95<{rb-g&}4  f2.{re-f}4  c2.{rb-g&}4
466: (t1)f2.{re-f}4  g2z100,,95,88f*4&g*32g8.g8
467:
468: (t3)o6z100,,98,104,108,112,110c1&  c4d4e-8f4<c8>  b-1
469: (t3)z100,95,90,85,80,100,90,102>'b-2<g''b-2<d' 'g2b-''f2a-'
470: (t3)'e-2g'f8e-4f8  >@u88'b-1<d'&  'b-2<d'<@u86'f2b-'
471:
472: (t4)@u98o3'a-1^1<c'  'b-1<d'&
473: (t4)'b-1<d'  'a-1^1<c'  @u80'c1^1e-gb-<d'
474:
475: (t5)o4r1r1  r2r8z100,88'b-<dg'rr4  'gb-<df'r2... r1r1r1  r1
476:
477:
478: (t6)o1z100,90,100a-4..a-r8a-4.  z100,90,100,90,88
479: (t6)b-4..b-r8b-4b-<d z100,90,100,90,80,90e-4..e-r8e-8.d>b-g
480: (t6)z100,90,100,80,90e-4..e-r8e-8.b-<@b0,-8191,12e-8@b0
481: (t6)d-4..d-r8d-^4>a-  z100,90,100<d-4..d-r8d-4.  z100,90
482: (t6)c4..cr8c4^>g  o2z100,80,90,95,100,90,80,90c4..gb-b<cg8c
>b8
483:
484: (t7)o5z100,90,80,85,100,90,82e-8.d8.>b-8<c8.e-8.d8
485: (t7)c2.z90,85>{b-<cr}4  z100,88,80d8.>g8.b-8^2
486: (t7)r1  z88,108,98o5|:2r8f8.g8.<c2>:|  |:2r8f8.g8.<d2>:|
487:
488: (t8)r1r1r1r1  z60,65,70,75,70,65,60
489: (t8)o4|:2{rrr'd-f''e-g''fa-''gb-''fa-''e-g'}4'fd-8'z65,60
490: (t8){fd-fd-fd-fd-fd- fd-fd-fd-fd-fd-}2r8 :|
491: (t8)z60,65,70,75,80,75,70,65,60,60
492: (t8)|:2{rrr'df''e-g''fa-''gb-''a-<c''gb-'}4{'fa-''e-g''df'&
}8
493: (t8)'d8f'z65,60{fdfdfdfdf dfdfdfdfd}2:|
494:
495:
496: (t10)z114,60,,80,110,60,,80,60,,100,80,110,60,,60
497: (t10)o3c+8>f+f+f+*0&df+f+f+ f+f+a+8 f+*0&df+f+f+
498: (t10)z80,60,,,80,110,60,,80,60,,100,80,110
499: (t10)|:3f+f+f+f+ f+*0&df+f+f+ f+f+a+8 f+*0&d@u60f+f+f+:|
500: (t10)z118,60,,,80,110,60,,80,60,,100,80,120
501: (t10)<a>f+f+f+ f+*0&df+f+f+ f+f+a+8 f+*0&f@u60f+f+f+
502: (t10)z80,60,,,80,110,60,,80,60,,100,80,120
503: (t10)|:2f+f+f+f+ f+*0&df+f+f+ f+f+a+8 f+*0&f@u60f+f+f+:|
504: (t10)z120,60,,,80,110,60,,80,115,,80,,120,60,80
505: (t10)<c+>f+f+f+ f+*0&df+f+f+ 'd<a'f+f+f+*0&<c> f+f+
506: (t10)z80,127,100f+*0&ga+
507:
508: (t11)w5w7w8w15w16w18o2@u127|:7c4..c8.c4.:|  c4..c8.c8.c8.
509:
510: (t14)o4z100,90,110|:4r1  r4.c+c+c2:|
511:
512: (t15)z70,75,80,75,70,85
513: (t15)o3r'a-<c''b-<d'<'ce-'r8>'b-<d'r'a-8.<c''b<d'r4
514: (t15)r'a-<c''b-<d'<'ce-'r8>'b-<d'r'a-8.<c'<'ce-'>r4  r1
515: (t15)z70,75,80,85,75,70,85
516: (t15)'b-<d''b-<d'<'ce-''df'r8>'b-<d'r'g8.b-''g<c'r4r1r1r1r1
517:
518: (t16)o4z70,50,70|:32a+8a+a+:|
```


▶資料請求ってお金がかかりますね。封書，封筒，切手(返信用270円も)，１枚出すたびに400円はかかりますから，10社に出したら4,000円，100社に出したら40,000円。就職活動するために働かねば……。　　今田　智宣(20)兵庫県

```
519:
520: (t18)z85,,105|:8o3bbf8 bbf8 bbf8 bbf8:|
521:
522: /------------------- D - toCODA + CODA ------------------
523:
524: (t1)o5z100,90,80,85,100,90,82
525: (t1)'g8.<e-''f8.<d''d8b-''e-8.<c''g8.<e-''f8<d'
526: (t1)'e-2.<c'z90,85{r'db-''e-<c'}4 z100,88,80'f8.<d'g8.'d8^2
b-'
527: (t1)z95,105,115,110,100,90,100,,78rde-fd>b-gb-&<c8c8
528: (t1)z108,98,,100,110,95{rb-g&}4  f2.{re-f}4
529: (t1)c2.z114,98,106,106{rb-g}4  b-*4&<c*140>{rb-<c}4
530: (t1)e-2&e-8z110,88g4.  <c1^1
531:
532: (t3)o6z100,,98,104,108,112,110c1&  c4d4e-8f4<c8>  b-1
533: (t3)z100,95,90,85,80,100,90,102>'b-2<g''g2<d'  'g2b-''f2a-'
534: (t3)'e-2g'f8e-4f8  @u98g1&  g1&  g1  r1
535:
536: (t4)@u98o2@d1'a-1<a-<c'  @u100b-1@d0  @u88o3'e-1b-1<d'&
537: (t4)'e-b-1<d'  'd-a-1^1<c'  @u80>'a-1^1^1^1<e-a-<c'
538:
539: (t5)o4r1r1  r2r8z100,88'b-<dg'rr4
540: (t5)'gb-<df'r2...  r1r1r1  r1r1r1
541:
542: (t6)o1z100,90,100,90,80a-4..a-r8a-4a-a z100,90b-4..b-r8b-4.
543: (t6)z100,90,100,90,80,90<e-4..e-r8e-8.d>b-g
544: (t6)z100,90,100,80,90e-4..e-r8e-8.b-<e-8
545: (t6)z100,90d-4..d-r8d-^4>a-  z100,90,100<d-4..d-r8d-4.
546: (t6)>a-4..a-r8a-4.  a-4..a-r8a-4.^1^1
547:
548: (t7)o5z100,90,80,85,100,90,82e-8.d8.>b-8<c8.e-8.d8
549: (t7)c2.z90,85>{b-<cr}4  z100,88,80d8.>g8.b-8^2  r1
550: (t7)z88,108,98o5|:3r8f8.g8.<c2>:| r8b-8.<c8.g2 @u105<c1& c1
551:
552: (t8)r1r1r1r1  z60,65,70,75,70,65,60
553: (t8)o4|:2{rrr'd-f''e-g''fa-''gb-''fa-''e-g'}4'fd-8'z65,60
554: (t8){fd-fd-fd-fd-fd- fd-fd-fd-fd-fd-}2r8 :|
555: (t8)z60,65,70,75,80,75,70,65,60,60
556: (t8)|:2{rrr'df''e-g''fa-''gb-''a-<c''gb-'}4{'fa-''e-g''df'&
}8
557: (t8)'d8f'z65,60{fdfdfdfdf dfdfdfdfd}2:|
558:
559:
560: (t10)z114,60,,121,110,60,,80,60,,100,80,110,60,,60
561: (t10)o3g8>f+f+<c+*0&>df+f+f+ f+f+a+8 f+*0&df+f+f+
562: (t10)z80,60,,,80,110,60,,80,60,,100,80,110
563: (t10)|:3f+f+f+f+ f+*0&df+f+f+ f+f+a+8 f+*0&d@u60f+f+f+:|
564: (t10)z114,60,40,80,110,60,,80,60,,108,80,110,60,,60
565: (t10)<a8>f+f+ f+*0&df+f+f+ f+f+a+8 f+*0&df+f+f+
566: (t10)z80,60,,,80,110,60,,80,60,,100,80,120
567: (t10)|:2f+f+f+f+ f+*0&df+f+f+ f+f+a+8 f+*0&f@u60f+f+f+:|
568: (t10)z80,60,,,80,110,60,,80,60,,80,60,,114
569: (t10)f+f+f+f+ f+*0&df+f+f+ f+f+f+8 f+f+<f8  z118a4
570: (t10)x$40,$10,$14,2{@u5c+c+c+@u10c+c+c+@u15c+c+c+@u20c+c+c+
@u25c+c+c+@u30c+c+c+@u35c+c+c+@u40c+c+c+c+@u50c+c+c+@u55c+c+c+@u
60c+c+c+@u65c+c+c+@u70c+c+c+c+@u60c+c+c+@u50c+c+c+@u40c+c+c+}2.
571: (t10){@u20bb@u30bb@u40bb@u50bb@u60bb@u70bb@u80bb@u90bb@u100
bb@u90bb@u80bb@u70bb@u60bb}2.
572: (t10)@u50{bbbb@u40bb@u20bbb}4x$40,$10,$14,0
573:
574: (t11)w13o2@u127|:8c4..c8.c4.:|  r1r1
575:
576: (t13)r*1728  o7@ar2..z100,90,80,70,60,50,40,30{fedc>bag}8@u
80
577:
578: (t14)o4z100,90,110|:4r1  r4.c+c+c2:|  r1r1
579:
580:
581: (t15)z70,75,80,75,70,85o3r'a-<c''b-<d'<'ce-'r8>'b-<d'r'a-8.
<c'
582: (t15)'b<d'r4 r'a-<c''b-<d'<'ce-'r8>'b-<d'r'a-8.<c'<'ce-'>r4
583: (t15)r1  z70,75,80,85,75,70,85
584: (t15)'b-<d''b-<d'<'ce-''df'r8>'b-<d'r'g8.b-''g<c'r4w
585:
586: (t16)o4z70,50,70|:40a+8a+a+:|
587:
588: (t18)z85,,95|:8o3bbf8 bbf8 bbf8 bbf8:|
589:
590: /------------------------ G ----------------------------
591: /譜面+2小節
592:
593: (t1)r1r1r1   r1r1r1r1  _48r2r8o4z68,78,88g8<c8g8
594: (t1)¥7z78,94,82,70,80,90,95,85d8.a8.f4.>a8<c8  d8.a8.f8^2
595: (t1)z90,80,70,60,88,,92,94,70,75,82,,78g8.g8.e8c8.>g&a8.<e
596: (t1)a4>g8<c4>b&<c@b0,-8191,24g4@b0  z80,94,88d8.a8.f8^2
597: (t1)z80,94,88d8.a8.f8^2  z95,90,72,84,80,70,64g8.g8.c8g8.g8
.c8
598: (t1)>a4r4..z70,75,80,85,90,90,80,60,92ga<cde  f8.d8.>a8<g2
599: (t1)z90,80,60,92f8.d8.>a8<g2
600: (t1)z95,90,72,84,80,70,64g8.g8.c8g8.g8.c8  >a4r2.
601:
602: (t6)o2z100,90,100,90,80d2...>a  <d2..>gg+  z100,90a2...e
603: (t6)a2..a8  z110,,90,100,90,80<@b-3415d*2&@b0d*178>a <d2..>
gg+
604: (t6)z100,90,80,108a2^8.aa<(a8.b),12  z100,90a2..>a8
605: (t6)¥7z100,90,100,90,80,100,90<d2...>a  <d2..>gg+  a2...<e
606: (t6)z100,,105,,110,,110g+*4&a*68a&b8.b&@b683,-8191,36b^4@b0
607: (t6)z100,90d2...>a  z100,90,100,80<d^2.d>gg+  z100,90a2...e
608: (t6)a2..a8  <d2...>a  <d1  >z100,90a2...e  a1
609:
610: (t2)|:2|:4o5{c<c>c<c>c<c}4:|  |:4o5{dfdfdf}4:|
611: (t2)|:8o4{a<g>a<g>a<g>}4:|:|  ¥7|:3|:4o5{c<c>c<c>c<c}4:|
612: (t2)|:4o5{dfdfdf}4:|  |:8o4{a<g>a<g>a<g>}4:|:|
613:
614: (t3)@u88o4|:2'cg'*768:|  ¥7|:3'cg'*768:|
615:
616: (t4)o2z98,94|:2'd<f'*384  'a<g<d'*384:|
617: (t4)¥7|:3'd<f'*384  'a<g<d'*384:|
618:
619: (t5)@1o6q8~11r1r1y$7e,1r1  z118,108,98,100,90,102,88e2d2
620: (t5)c2.r4  r2{r>bg}2  a4.e8^2  r2.z100,80,95,95{rrd>g<e&e}4
621: (t5)¥7z100,90,80,100,92,90,100,110,92,,92g2.{rfe}4
622: (t5)f4.<c4.>{rb<c}4  {d>g&g&}2g4r4
623: (t5)z90,110,78,98,90,100,110,78,98r4{rdg}4d+*4e*44>b8<cd
624: (t5)>b*4<c*140r4  z90,,100,110,78,98r2r8g8&{ga<c}4
625: (t5)>g*4a*140r4  r2r8z80,90,,100,110,120,127,100,88,,80,100
b4.
626: (t5)<c2&{cde}2  {fe>a&}2a4{r<cg}4  z79,92,85d*4e*44r>a4r4.r
627: (t5)r2z78,83,78,100,88g8d8a*4g*20a8
628:
629:
630: (t10)z80,60,,60,100,60,60o2f+4f+f+8.f+*0&c+4f+f+8.
631: (t10)z80,105,80,60,80,40,100,60,80,60,80
632: (t10)f+8a+8f+8f+f+f+*0&c+4f+f+f+f+
633: (t10)z40,60,,60,100,60,60f+4f+f+8.f+*0&c+4f+f+8.
634: (t10)z80,105,80,60,80,40,100,60,80,60,80
635: (t10)f+8a+8f+8f+f+f+*0&c+4f+f+f+f+
636: (t10)z40,60,,60,100,60,60f+4f+f+8.f+*0&c+4f+f+8.
637: (t10)z80,105,80,60,,80,100,60,80,60,80
638: (t10)f+8a+8f+8f+f+f+*0&c+4f+f+f+f+
639: (t10)z40,60,,60,100,60,60f+4f+f+8.f+*0&c+4f+f+8.
640: (t10)z80,60,100,80,60,80,80,100,60,80,60,80,60,80
641: (t10)f+f+a+8f+f+a+8f+*0&c+f+f+8f+f+f+f+
642: (t10)¥7z80,60,80,60,80,60,100,60,80,80f+f+f+8f+f+8f+c+f+f+8
f+4
643: (t10)z80,60,100,80,60,80,80,100,80,60,80,60,80
644: (t10)f+f+a+8f+8f+f+f+*0&c+8f+8f+f+f+f+
645: (t10)z80,60,80,60,80,60,100,60,80,80
646: (t10)f+f+f+8f+f+8f+c+f+f+8f+4
647: (t10)z80,60,100,80,60,,,100,60,,80,60,,100,100
648: (t10)f+f+a+8f+f+f+f+c+f+f+f+f+f+a+a+
649: (t10)z81,60,,80,,60,,100,60,80,60,80
650: (t10)f+f+f+f+f+8f+f+c+f+f+8f+f+8.
651: (t10)z80,60,100,80,60,80,80,100,80,60,80,60,80
652: (t10)f+f+a+8f+8f+f+f+*0&c+8f+8f+f+f+f+
653: (t10)z80,60,80,60,80,60,80,100,80f+f+f+8f+f+f+f+f+*0&c+4f+4
654: (t10)z81,60,,80,,60,,100,60,80,60,80
655; (t10)|:5f+f+f+f+8f+f+c+f+f+8f+f+8.:|
656:
657: (t11)w2w12z50,127,127
658: (t11)|:4o5a*0&o2c1  c1:|  ¥7|:6o5a*0&o2c1  c1:|
659:
660: (t12)|:4@p27r2rb8rg8r8  @p100r2.g8r8:|
661: (t12)¥7|:6@p27r2rb8rg8r8  @p100r2.g8r8:|
662:
663: (t13)r*1536  ¥7r*1344  o8@ar2@a127
664: (t13){fedc>bagfedc>bagfedc>bagfedc>bagfedc>bagfedc>bagfedc>
bagfedc>ba}2^1  r1
665:
666: (t14)z78,,98o5|:4r2..a+a+  r2.d+4:|  ¥7|:6r2..a+a+ r2.d+4:|
667:
668: (p)
```

リスト5 THE SUMMER OF '68用カウンタ表示

```
 1:00005280 00000000   2:00003780 00000000   3:00005280 00000000   4:00005280 00000000
 5:00002580 00000000   6:00004200 00000000   7:00001B00 00000000   8:00001800 00000000
 9:00000C60 00000000  10:00004200 00000000  11:00005280 00000000  12:00003780 00000000
13:00003D80 00000000  14:00004200 00000000  15:00001200 00000000  16:00001980 00000000
17:00000600 00000000  18:00000C00 00000000
```

リスト6 トゥインクルトゥインクル

日本音楽著作権協会(出)許諾第9570183-501号

```
================  TWINKLE.ZMS  ===================
 1: / twinkle.zms
 2: ////////////////////////////////////////////////////////////
 3: /
 4: /     トゥインクル トゥインクル / Wink
 5: /
 6: /     作詞:秋元 康
 7: /
 8: /     作曲・編曲:ジェイムス下地
 9: /
10: ////////////////////////////////////////////////////////////
11: /
12: / 製作期間 95/02/03 ~ 95/02/09
13:
14: .comment Twinkle Twinkle / Wink for SC-88 by M.Hasunuma
15:
16: (i)(b1)
17: (m01,5000)(amidi11,01) (m02,5000)(amidi11,02)
18: (m03,5000)(amidi01,03) (m04,5000)(amidi02,04)
19: (m05,5000)(amidi07,05) (m06,5000)(amidi04,06)
20: (m07,5000)(amidi04,07) (m08,5000)(amidi05,08)
21: (m09,5000)(amidi05,09) (m10,5000)(amidi06,10)
22: (m11,5000)(amidi10,11) (m12,5000)(amidi10,12)
23: (m13,5000)(amidi15,13) (m14,5000)(amidi16,14)
```

▶4月から会社員。4月から東京都民。念願の技術屋になった。給料日が楽しみだなぁ。
雨宮 光児(20)東京都

```
 24: (m15,5000)(amidi03,15) (m16,5000)(amidi08,16)
 25: (m17,5000)(amidi09,17) (m18,5000)(amidi12,18)
 26: (m19,5000)(amidi13,19) (m20,5000)(amidi14,20)
 27:
 28: /=====================================================
 29: /SC-88 system setup
 30:
 31: .roland_exclusive $10,$42 {0,0,127,0}
 32:                     / SC-88 MODE 1
 33:
 34: .sc55_reverb {  0  / reverb macro
 35:                 3      / reverb character
 36:                 2      / reverb pre-lpf
 37:               120      / reverb level
 38:                70      / reverb time
 39:                 0      / reverb delay feedback
 40:                 0}     / reverb send level to chorus
 41:
 42: /=====================================================
 43:
 44: (t01,02,03,04,05,06,07,08) @i$41,$10,$42
 45: (t09,10,11,12,13,14,15,16) @i$41,$10,$42
 46: (t17,18,19,20)             @i$41,$10,$42
 47:
 48: /=====================================================
 49: (t1)
 50: t96
 51: n11x$40,$1a,$14,1,2
 52: i0,2@2@v110u100p3r@r1                     / drum
 53: r2
 54: r1r1 r1r1 r1r1 r1r2
 55: L8rrr t132r4.
 56:
 57: o2q4L4u110
 58: |:4c|'ce':|'c8.e'u-25'c16e'u
 59: |:4c|'ce':|'c16e'u-25|:3'c16e':|u
 60: |:4c|'ce':|'c8.e'u-25'c16e'u
 61: |:3c'ce':|L16ceer'ce'u-25|:3'ce':|uL4
 62:
 63: |: o2L4u110 |:4
 64: |:4c|'ce':|'c8.e'u-25'c16e'u
 65: |:4c|'ce':|'c16e'u-25|:3'c16e':|u
 66: |:4c|'ce':|'c8.e'u-25'c16e'u
 67: |:3c'ce':|L16ceer'ce'u-25|:3'ce':|uL4
 68: :| | r2.'c16e'u-25|:3'c16e':|u
 69: :|
 70:
 71: |:4c|'ce':|'c8.e'u-25'c16e'u
 72: |:4c|'ce':|'c16e'u-25|:3'c16e':|u
 73: |:4c|'ce':|'c8.e'u-25'c16e'u
 74: |:3c'ce':|L16ceer'ce'u-25|:3'ce':|uL4
 75:
 76: r1r1 r1r1 r1r1 r1r1 cccccccc
 77: cccc|:4c16u-20c8u:|{cc}cc L16ceer'ce'u-20|:3'ce':|uL4
 78:
 79: |:4
 80: |:4c|'ce':|'c8.e'u-25'c16e'u
 81: |:4c|'ce':|'c16e'u-25|:3'c16e':|u
 82: |:4c|'ce':|'c8.e'u-25'c16e'u
 83: |:3c'ce':|L16ceer'ce'u-25||:3'ce':|uL4
 84: :|r8.
 85:
 86: /=====================================================
 87: (t2)
 88: n11r@r1                                   / tamb , c.cym
 89: r2
 90: r1r1 r1r1 r1r1 r1r2
 91: L8rrrr4.
 92:
 93: o3L16q4|:|:3u85c+u65f+f+r:| u65|:13rf+f+r:|:|
 94: |:o3L16q4
 95: |:u85c+*0u65 |:32rf+f+r:|:|
 96: |:4|:3u85c+u65f+f+r:| u65|:13rf+f+r:|:|
 97: | r1 :|
 98: u85c+*0u65|:32rf+f+r:|
 99: u85c+1u65r1 r1r1 r1r1 r1r1
100: u85c+1u65r1 r1r1 r1
101: |:8|:3u85c+u65f+f+r:| u65|:13rf+f+r:|:|
102:
103: /=====================================================
104: (t3)
105: n1i0,2@37@v100u100p3@e10r           / bass
106: y126,1 r2 r1r1 r1r1 r1r1 r1r2
107: o2@v90u120L8rrrq6>b-<c+f
108:
109: o2@v107u120q6
110: L4e-e-a-a-ffb-b- ffa-a-b-b-b-b-
111: e-e-a-a-ffb-b- <cccc>ffff
112:
113: |:
114: o2L4q4
115: b-b-b-b-aaaa a-a-a-a-e-e-e-e-
116: e-e-a-a-c+c+>b-b-< ccccffff
117:
118: b-b-b-b-aaaa a-a-a-a-e-e-e-e-
119: e-e-a-a-c+c+>b-b- f+f+ffb-b-b-b-<
120:
121: |:>b-b-a-a-ffb-b- f+f+a-a-<c+c+c+c+
122:   >b-b-a-a-ffb-b- f+f+ffb-b-b-b-<:|
123:
124: |r1:|
125:
126: >b-b-a-a-ffb-b- f+f+a-a-<c+c+c+c+
127: >f+f+a-a-ffb-b- e-e-ffb-b-b-b-<
128:
129: r1r1 r1r1 r1r1 r1r1
130: @v70>c+c+e-e-fff+f+<
131: @v85>ffff|:4f16f8:|f8f8fff+r<@v107
132:
133: |:4>f+f+aaf+f+bb ggaa<dddd
134:    >f+f+aaf+f+bb ggf+f+bbbb<:|
135:
136: /=====================================================
137: (t4)
138: n2i0,2@10@v100@u110p3r                      / glocken
139: o5L8rb-<c+f b-^b-^a-2 r1 r1 r1
140:             b-^b-^a-2 r1 r1 r2r2.
141:
142: r1r1 r1r1 r1r1 r1i4@99r2.L8o5@v100u100rc
143:
144: |:
145: c+.>b-16&b-2. <r2.rc
146: c+.>b-16&b-2. <r1
147: r1r1 r1r2.rc
148:
149: c+.>b-16&b-2. <r2.rc
150: c+.>b-16&b-2<b-f e-1
151: r1r1 r1r1
152:
153: u110|:4r2.rf+ff+a-b-^2 r1r1 :| |r1:|
154:
155: r1r1 r1r1 r1r1 r1r1
156:
157: o5L8b-4.c+^2r1 r1r1
158: f+4.>b-^2<r1 r1r1
159: r.c.c+e-.c+.e-f1 r1r1r1
160:
161: u110
162: |:8r2.rgf+gab^2 r1r1 :|
163:
164: /=====================================================
165: (t5)
166: n7i8,2@56@v100u100p3r                       / hit
167: r2@y1,$66,44@y1,$63,54@y1,$64,94
168: r1r1 r1r1 r1r1 r1r2
169:
170: o4u115L8@e30rrrq6b-<c+f@e80
171: q8b-4b-4a-2r1 r1r1
172: q8b-4b-4a-2r1 r1
173: L16@e30rq3fff |:q5f8q3f:|q5f8q8@e80f4
174:
175: |:r*3072
176: |:4q8o5b-4b-4a-2r1 r1r1:| |r1:|
177:
178: r1r1 r1r1 r1r1 r1
179: @e30rq3b-b-b- |:q5b-8q3b-:|q5b-8q8@e80b-4
180:
181: r1r1 r1r1 r1r1 r1r1 r1r1 r1
182: @e30|:4q7f16q5f8:|f8f8q6L4fff+r
183:
184: |:8
185: @e80q8o5b4b4a2r1 r1|r1:|
186: @e30L16rq3bbb|:q5b8q3b:|q5b8 q8b4
187:
188: /=====================================================
189: (t6)
190: n4i0,2@26@v100@u100p1r@g12          / guitar (L)
191: r2
192: @v50o4L8u100'fc+'>f<c+e->a-b-^<e-
193: f>f^b-<c+>f^q4fq8
194: re-fb- >a-<e-<u+10a-q2u-20cq8u
195: c+>f^<a-f>f^u-10fu<
196:
197: 'fc+'>f<c+e->a-b-^<e-
198: f>f^b-<c+>f^q4fq8
199: <f+4c+q2f+q8f^>a-<c
200: c+>fb-<c+f2^4
201:
202: r1r1 r1r1 r1r1 r1r1
203:
204: |:r*5952 |r1 i9@26o2~20'e-1b-<e-',2_ :|
205: i0@26r1
206:
207: p3o4@v70u102|:3q7f4.q3f:|q7c+2
208: f4.q4fq7f4.q4f L16rrfrra-rrf2
209: r1r1 r2.L8rq6<c> r1
210:
211: p1@v60u100
212: o4L8q7e-c+f+f^2 e-cfa-4c+f4
213: >fa-b-<c+^2 >fa-b-<f^2
214: f+>b-<e-f+^2 f>fa<c4c>b-<c
215: >b-a<c>a^2 a-4.g^2
216: o4|:'cfa-2',4:|'>b-<c+f+1'
217: f>f<cf4c>f<c^1r1
218:
219: /=====================================================
220: (t7)
221: n4@u100p1r                              / 〃
222: r2
223: u105o4L4rrra-r>a-<rf
224: >b-2a-4<q5c4q8 >ra-ra-<
225:
226: rrra-r>a-<rf
227: r8>f+<q3c+8q8r8>f8r
228: >b-8&b-4.r8<u-20f8b-2
229:
230: /=====================================================
231: (t8)
232: n5i0,2@25@v100@u100p2r                      / guitar (R)
233: r2
234: @v70o4L8u90f>f<c+e->a-b-^<q4e-q8
235: f>f^b-<c+>f^q4fq8
236: re-fb- >a-<e-<u+10a-q2u-20cq8u
237: c+>f^<a-f>f^u-10fu<
238:
239: f>f<c+e->a-b-^<q4e-q8
240: f>f^b-<c+>f^q4fq8
241: <f+^c+q4f+q8f^>a-<c
```


▶おそらく私の書き間違いでしょう。私の所有機種は「X68000CompactXVI」ではなく「X68030Compact」です。これをいいたくて……。　山下　守政(19)兵庫県
(担当者：すみませんでした)

```
242: c+>fb-<c+f2^4
243:
244: r1r1 r1r1 r1r1 r1r1
245:
246: |:r*5952|r1 o4~20'ce-a-1',3_:|r1
247:
248: r1r1 r1r1 r1r1 r1r1
249:
250: @v60u100
251: o4L8q7|:e-c+f+f^2 :|
252: >fa-b-<c+^2 >fa-b-<f^2
253: f+>b-<e-f+^2 f>fa<c4>a<c4
254: b-4.<c^2c+4.>b-^2
255: o4|:'cfa-2',3:|'>b-<c+f+1'
256: f>f<cf4c>f<c^1r1
257:
258: /======================================================
259: (t9)
260: n5@u100p2r                          / "
261: r2
262: u98o4L4rrra-r>a-<rf
263: >b-2a-4<c4 >ra-ra-<
264:
265: rrra-r>a-<rf
266: r8>f+<q3c+8q8r8>f8r
267: >b-8&b-4.r8<u-20f8b-2
268:
269: /======================================================
270: (t10)
271: n6i0,1@25@v100u100p3r@g12           / guiter (C)
272: r2
273: @v90u80o4L4c+>b->a-<q6cq8 c2>b-2
274: b-2q6a-4<c4q8 c+1
275:
276: >b-<b->a-<q6cq8 c2>b-2
277: b-2a-2<f1 r4
278:
279: r1r1 r1r1 r1r1 r1r1
280:
281: |:r*5952|r1 o2b-1 :|
282: i9,2@26@y1,$21,78@y1,$64,74r2.u117r8.o3@d1c+48f+24
283: o4L8c+.>@d0f+.<q4c+ q8@d1e-.>a-.<q4@d0e-
284: q8@d1c.>f.<q4@d0c q8@d1c.>b-.q7@d0<u-10cu
285: @d1c+.>f+.<q4@d0c+ q8@d1e-.>a-.<q4@d0e-q8
286: r@b2732,1366,6a-16&@b1366,0,0a-16&@b0a-16
287: q4u-10'a-16e-'ur q7c.>f.<q4c
288: L16q7@b-683,0,6c+&@b0c+f+q6b-r<c+r>b-L8<q8c.>q5a-.<q5c>
289: L16q7cr@d1a-<c8>q4@d0a-rq6u-10'b-fc+'
290: rq8'b-8fc+'u-30'b-fc+'u+30'b-8fc+'u-20q3fru
291: q7'b-f+'u-35'b-f+'u+10'b-f+'u'b-f+'
292: 'b-f+''b-f+'r'a8f''a8f'u-25'g8.f'u<'cf'r>
293: {r<c+c>b-a-f+f e-c+e-}2.
294: |:3@b1366e-&@b2049e-&@b1366e-&@b0e-_15:|r2@v90r1
295:
296: r1r1 r1r1 r1r1 r1r1
297: r1r1r1
298:
299: r1r1 r1r1 r1r1 r1r1
300: r1r1 r1@y1,$64,48r2...o2u106@q1f+48<c+24
301: o4L8f+4f+4e.e.{rf+g}f+2f+.f+.f+
302: g2e.f+.q3b- rbbrbrrr
303:
304: o5q7L16f+f+q4f+erdrc+r2 r4rrq7ed e.d.c+ >ba.g.
305: f+.e.def+r8r2 <rrf+rre8rd8rc+8rd>u-20f+u
306: @m50f+1@m q6<f+rre8rdr d8rc+rr>br
307: u-20@q2f+4f+4L8e.u@m30b.<q5c+@q2d2&
308: ;$cc,10                             / all ch. fadeout
309: d2@m
310: q6L16o4<f+>rru'f+<e'rr'f+<e'r r'e<e'rr'e<f+'rr'e<e'
311: rr'f+<d'rr'f+<c+'rr 'f+<d'rr'a<e'rr'f+<d'r
312: r'f+<c+'rr'f+<d'rr<e> rr'f+<d'rr'f+<c+'au-20f+u
313: <f+rrerrdr c+rdrc+>bu-20aub
314: rargf+edef+4r4 r2reu-20aub <du-20c+>b<ud
315: c+.>b.agf+.e. def+a8.b-8 r8b4.^2
316:
317: /======================================================
318: (t11)
319: n10i0,2@9@v100u100p3r@e100@r1                / hihat
320: @y$18,75,65 @y$1d,75,127 @y$1c,83,64
321: o4L8q4u40rra+r
322: |:a+ra+ra+<u80d+>|u40a+r:|u70<b*0&>u40a+r
323: |:a+ra+ra+<u80d+> u40a+r:|
324:
325: |:a+ra+ra+<u80d+>|u40a+r:|u70<b*0&>u40a+r
326: |:a+ra+ra+|<u80d+> u40a+r:|rrr r4
327:
328: o2L16u80@e50
329: |:8f+f+a+8 f+f+a+8 f+'f+<<<d+'a+8 f+f+a+8 :|
330:
331: |: |:8
332: |:4f+f+a+8 f+f+a+8 f+'f+<<<d+'a+8 f+f+a+8 :|
333: :| | r1 :|
334:
335: |:
336: |:4f+f+a+8 f+f+a+8 f+'f+<<<d+'a+8 f+f+a+8 :|:|
337:
338: o4|:u60b-4u40b-4r4b-4:|u60b-4r2.b-4r2.
339: |:b-4r2.:| |:b-4r8b-8r2:| b-2b-2b-2b-2 b-1r1r1
340:
341: o2u80
342: |:32f+f+a+8 f+f+a+8 f+'f+<<<d+'a+8 f+f+a+8 :|
343:
344: /======================================================
345: (t12)
346: n10r@r1                                         / maracas
347: r2
348: @y$18,60,62 @y$1c,60,127 @y$1d,60,20
349: r1r1 r1r1 r1r1 r1r2
350: r4.r4. @y$1a,70,75
351:
352: o4L16u80q4
353: |:
354: r2r8ccrra+a+ r2r8ccra+a+a+ r2r8ccrra+a+ r2r8cca+a+a+a+
355: :|
356:
357: |:|:8
358: r2r8ccrra+a+ r2r8ccra+a+a+ r2r8ccrra+a+ r2r8cca+a+a+a+
359: :| |r1:|
360:
361: |:
362: r2r8ccrra+a+ r2r8ccra+a+a+ r2r8ccrra+a+ r2r8cca+a+a+a+
363: :|
364:
365: r1r1 r1r2o5b2 r1r1 r1r2b2 r1r1 r1r1r1
366:
367: o4u80
368: |:8
369: r2r8ccrra+a+ r2r8ccra+a+a+ r2r8ccrra+a+ r2r8cca+a+a+a+
370: :|
371:
372: /======================================================
373: (t13)
374: n15i0,2@27@v100u100p3@g12r          / chorus
375: m,1@c7s,3
376: r2@y1,$63,64@y1,$64,64@e20,10
377: r1r1 r1r1
378: @v70u100@m12
379: o5L8@b-683,-342c+16&@b0c+8.c+4 q6cq8cr>b-
380: q7a-a-q5a-q7a- q8b-4r4
381: r1r1 r4 r1r1 r1r1 i0@27
382: @v60u110o4@b-683,0'b-16c+<b-'&@b0'b-.c+<b-''b-4c+<b-'
383: q5'a-c<a-'q8'a-4.c<a-' r1r1r1
384:
385: |:
386: r1r1r1r1 i0@27@v65u100@y1,$63,64@y1,$64,64
387: r1r1r1r2.|:ro5q4'ec>e'
388: q6@b-683,0'f16c+>f'&@b0'fc+>f'q7'c+.>b-c+'r r2r2.:|r
389: i0@55@v80u120 @y1,$63,44@y1,$64,44r
390: o4q7@b-683,0'f+16>b-'&@b0'f+4..>b-'
391: @b-683,0'f+16c'&@b0'f+4..c'
392: 'f2>a-'q5@b-683,0'f16>b-'&@b0'f4..>b-'
393: q7@b-683,0'f+16e->f+'&@b0'f+4..e->f+''f2e->a'q6'c+2>b-f'
394: i0@27@v60u102r@y1,$63,64@y1,$64,64o5q8'c+>c+'q4'f>f''a->f'
395:
396: |:
397: q7o6@b-683,0'c+16>c+>f+'&@b0'c+.>c+>f+''c+4>c+>f+'
398: q5'c>c>e-''c4>c>e-' >q5'b->b-c+'
399: q7'a->a-c''a->a-c+'q5'b->b-e-'<'c+>c+>f' >q8'b-4>b-c+'r
400: i0@55@v80u120 @y1,$63,44@y1,$64,44r
401: o4q7@b-683,0'b-16c+'&@b0'b-4..c+'
402: 'a-4.e-'&@b0,-1366'a-e-'@b0'f2.>a-'r4
403: i0@27@v60u102@y1,$63,64@y1,$64,64
404:
405: q7o6@b-683,0'c+16>c+>f+'&@b0'c+.>c+>f+''c+4>c+>f+'
406: q5'c>c>e-''c4>c>e-' >q5'b->b-c+'
407: q7'a->a-c''a->a-c+'q5'b->b-e-'<'c+>c+>f' >q8'b-4>b-c+'r
408: i0@55@v80u120 @y1,$63,44@y1,$64,44r
409: o4q7@b-683,0'b-16c+'&@b0'b-4..c+''a2c'|'b-2c+'r16
410: i0@27@v60u102r16@y1,$63,64@y1,$64,64
411: o5q8'c+>c+'q4'f>f''a->f'
412: :| o4q8'b-2.c+'|ri0@53@y1,$20,67@y1,$21,78r
413:
414: o5@c,,80@s,6@a40'a-4e-c>a-e-c'&
415: @c,0,100@s,30@a-60'a-2.e-c>a-e-c'@a
416: @y1,$20,64@y1,$21,64@v65
417: :|r4
418:
419: r*3072
420: i0@27@v60u102
421: r1o5L8r4q7@b-683,0b-32&@b0b-16.q5b-
422: q7b-.b-16&@b0,-342b- @b0q4f+
423: q7b-4.a^2^1r2
424: i0@27@v60u102r@y1,$63,64@y1,$64,64
425: o5q8'd>bd'q4'f+d>f+''bd>b'
426:
427: |:3
428: o6q7'd4>f+d''d4>f+d'q6'c+>ec+''c+4>ec+'
429: >q5'bd>b'q7'ac+>a''bad>a'q5<'c+>ae>a''d>af+>a'>q6'b4d>b'r
430: i0@55@v80u120 @y1,$63,44@y1,$64,44r
431: o4q7@b-683,0'b16d'&@b0'b4..d'
432: 'a4.e'&@b0,-1366'ae'@b0'f+2.>a'r
433: i0@27@v60u102@y1,$63,64@y1,$64,64r
434: o6q7'd4>f+d''d4>f+d'q6'c+>ec+''c+4>ec+'
435: >q5'bd>b'q7'ac+>a''bad>a'q5<'c+>ae>a''d>af+>a'>q6'b4d>b'r
436: i0@55@v80u120 @y1,$63,44@y1,$64,44r
437: o4q7@b-683,0'b16d'&@b0'b4..d''b-2c+''b2d'r16
438: i0@27@v60u102r16@y1,$63,64@y1,$64,64
439: o5q8'd>bd'q4'f+d>f+''bd>b'
440: :|
441: o6q7'd4>f+d''d4>f+d'q6'c+>ec+''c+4>ec+'r1
442: i0@55@v80u120 @y1,$63,44@y1,$64,44r
443: o4q7@b-683,0'b16d'&@b0'b4..d'
444: 'a4.e'&@b0,-1366'ae'@b0'f+2.>a'r
445: i0@27@v60u102@y1,$63,64@y1,$64,64r
446: o6q7'd4>f+d''d4>f+d'q6'c+>ec+''c+4>ec+'r1
447: i0@55@v80u120 @y1,$63,44@y1,$64,44r
448: o4q7@b-683,0'b16d'&@b0'b4..d''b-2c+''b2d'r2
449:
450: /======================================================
451: (t14)
452: n16                                             / vocal
453: i8,2@6@y1,$20,68@y1,$21,70
454: /i0,2@67
455: @v94u100p3@g12@e45,50r
456: @y1,$63,67 @y1,$64,74 @y1,$66,68
457: @m10o5L8r>b-<q7c+f
458: q7@b-683,0b-16&@b0b-.b-4q6a-q7a-4f+
459: ff+a-b-u-10c+4urq6c+q7
```

▶ついに出ましたね，PD。以前から目をつけていた製品なので，ぜひ手に入れたいところです。その前に受験だ……。　今井　佑(18)東京都

```
460: >b-4<c+c+c>b-q4a-<q7@b-683c+&@b0
461: c+cc+e-u-15@b-683,0f16&@b0f4^16ru
462: @b-683,0b-16&@b0b-.b-4a-a-4f+
463: ff+a-b-u-10c+4urq6c+
464: >b-4<q7@b-683,0c+16&@b0c+.q6c
465: >q7b-<c@b-1366c16.&@b-1366,0c32&@b0c>u-15b-2..r4u
466:
467: r1r1 r1r1 r1r1 r1u110o5L8r2.rq3c
468:
469: q7@b-683,0,3c+16&@b0c+8>b-16&b-rq7b-b-<c+q5e-
470: q7@b-683,0,5f16&@b0f@b-683,0f+16&@b0f+f&@m70f4rq3@m10c
471: q7@b-683,0,3c+16&@b0c+>q8b-.r
472: b-q3b-<q7c+e- @b-683,0,0f16&@b0f16e-f@m70e-4.r@m10q4c+
473:
474: q7c4e-4 rq4cq7c>a-q5a-<q7c+e-f^r
475: @b683,1366c+32&@b1366c+16.&@b0c+
476: q4cq7c4r @b-683,0c16&@b0c16cq5e-f+
477: q7@b-683f+32&@b0f+^32f16&f4&@m70f4@m10rq5c
478: |:q7@b-683,0c+16&@b0c+>b-.r b-u-15b-u<
479: |q5c+q7e- q5f.q7f+.f&@m70f4r@m10 q5c:|
480: b-f e-^24@b-1366,0,6f12&@b0fe-&@m70e-4@m10r q5c+
481: q7c4@m,,,50e-4@m10 rq4c>q7b-q6a-
482: <q7f^24e-^12fq6@m50c+4@m10 q7c>b-&
483: q2b-r<q7cq5c+q7@b-683,0,0e-16&@b0e-16c+c>q5a<
484: q7@b-683,0c16&@b0c16^24>b-12&@m70q6b-4@m10 o4rq7b-<q4c+f
485:
486: q7@b-1366,0b-16&@b0b-.b-4q5a-q6a-4q5f+ q7fq5f+a-b- q7c+4rc
487: >b-4<c+cr>a-r<@b-683,0,12c+&@b0c+cq5c+q7e-
488: @b-683,0,0f16&@b0@m40f.@m10r4
489: q7@b-1366,0b-16&@b0b-.b-4q5a-q6a-4q5f+ q7fq5f+a-b- q7c+4rc
490: >b-4<c+4q5c>q7b-<c
491: @b-1366,0,12c&@b0c>@m40b-4.@m10rb-<q4c+f
492: q7@b-1366,0b-16&@b0b-.b-4q5a-q6a-4q5f+ q7fq5f+a-b- q7c+4rc
493: >b-4<q6c+q5c+q4c>b-a-<@b-683,0,12c+&@b0c+cq7c+e-
494: @b-683,0,0f16&@b0@m40f.@m10r4
495: q7@b-1366,0b-16&@b0b-.b-4q5a-q6a-4q5f+ q7fq5f+a-b- q7c+4rc
496: >b-4<c+u-10c+uq5c>q7b-<c @b-1366,0,12c&@b0c>@m40b-2@m10r4.
497:
498: u110o5L8r2.rq3c
499: q7@b-683,0,3c+16&@b0c+8>b-16&b-rq7b-q5b-<c+q7e-
500: q7f.q6f+.q7f&@m70f4rq3@m10c
501: q7@b-683,0,3c+16&@b0c+>q8b-.r
502: b-q3b-<q7c+e- @b-683,0,0f16&@b0f16e-f@m70e-4.r@m10q4c+
503: q7c4e-4 rq4cq7c>a-q5<c+q7c+e-f^r @b-683,0c+32&@b0c+16.@b0c
504: q7c4r4 @b-683,0c16&@b0c16ce-q5f+ q7f+f&@m70f4@m10r4rq5c
505: |:q7@b-683,0c+16&@b0c+>b-.r b-b-<
506: |q5c+q6e- q5f.q7f+.f&@m70f4r@m10 q5c:|
507: q5b-q6f e-^24@b-1366,0,6f12&@b0fe-&@m70e-4@m10r q5c+
508: q7c4@m50e-4@m10 rq4cq7c>q8a-
509: <q3fq4e-q7e-f16c+16&q8@m50c+@m10r q7c>b-
510: rr<q7cc+q6@b-683,0,0e-32&@b0e-16r32q5c+q7c>a<
511: q7@b-683,0c16&@b0c16^24>b-12&@m70q6b-4@m10 o4rq7b-<q4c+f
512:
513: q7@b-1366,0b-16&@b0b-.b-4q5a-q6a-4q5f+ q7fq5f+a-b- q7c+4rc
514: >b-4<q6c+q5c+q4c>b-a-<@b-683,0,12c+&@b0c+cq7c+e-
515: @b-683,0,0f16&@b0@m40f.@m10r4
516: q7@b-1366,0b-16&@b0b-.b-4q5a-q6a-4q5f+ q7fq5f+a-b- q7c+4rc
517: >b-4<c+4q5c>q7b-<c
518: @b-1366,0,12c&@b0c>@m40b-4.@m10rb-<q4c+f
519: q7@b-1366,0b-16&@b0b-.b-4q5a-q6a-4q5f+ q7ff+a-b- c+4rq5c
520: q7>b-4<c+c+c>b-q5a-<@b-683,0,12c+&@b0c+q7cc+e-
521: @b-683,0,0f16&@b0@m40f.@m10r4
522: q7@b-1366,0b-16&@b0b-.b-4q5a-q6a-4q5f+ q7fq5f+a-b- q7c+4rc
523: >b-4<c+4q5c>q7b-<c @b-1366,0,12c&@b0c>@m40b-2@m10r4.
524:
525: r1r1 r1r1 r1r1 r1r1
526:
527: @v94u110@y1,$64,94
528: o5L8q8b-4.c+4.@b-1366,0,18b-&@b0b-
529: a-4f+4q4fq8fq4e-q8@b-1366,0a-&@b0a-1&@m50a-2.@m10r4
530: q7@b-683,0,16f+&@b0f+4f+4.@b-1366,0a-&@b0a-
531: @b-683,0,10f&@b0fe-4c+4c4
532: @b-1366,0,16e-&@b0e-c+@b-683,0,6c+&@b0c+&@m50c+4.@m10r2.
533: >b-<q5c
534: q7@b-683,0,0c+16&@b0c+q4c.c+ q7@b-683,0e-16&@b0e-c+.q4e-
535: q6f4 c+q7c+ @b-683,0f+16&@b0f+ @b683,0f32&@b0f^32 q4e-
536: q7@b-683,0,12f+&@b0f+f2.&@m70f4.@m10r2r8
537: @y1,$64,74r2r>b<q4df+
538:
539: |:
540: q7@b-1366,0,0b16&@b0b.b4q5aq6a4q5g q7f+q5gab q7d4rc+
541: >b4<q6dq5dq4c+>ba<@b-683,0,12d&@b0dc+q7de
542: @b-683,0,0f+16&@b0@m40f+.@m10r4
543: q7@b-1366,0,0b16&@b0b.b4q5aq6a4q5g q7f+q5gab q7d4rc+
544: >b4<d4q5c+>q7b<c+
545: @b-1366,0,12c+&@b0c+>@m40b4.@m10rb<q4df+ |
546: q7@b-1366,0,0b16&@b0b.b4q5aq6a4q5g q7f+gab d4rq5c+
547: q7>b4<ddc+>bq5a<@b-683,0,12d&@b0dq7c+de
548: @b-683,0,0f+16&@b0@m40f+.@m10r4
549: q7@b-1366,0,0b16&@b0b.b4q5aq6a4q5g q7f+q5gab q7d4rc+
550: >b4<d4q5c+>q7b<c+ @b-1366,0,12c+&@b0c+>@m40b4.@m10rb<q4df+
551: :|
552:
553: |:q7@b-1366,0,0b16&@b0b.b4q5aq8a4r r1r1r1:|
554:
555: /=====================================================
556: (t15)
557: n3i1,2@49@v100@u100p3r                      / strings
558: r2
559: o5@v80u40@q2
560: 'f2>b-''e-2>a-' 'e-2>a-''c+2>f'
561: 'f2>b-''e-2>a->e-' 'c+2>f>c+''f2>b->f'
562: 'f2>b-''e-2>a-' 'e-2>a-''c+2>f'
563: 'f2>b-''e-2>a-' 'c+>f1^4'
564:
565: u75'f2>b-''e-2>a-' 'e-2>a-''c+2>f'
566: 'f2>b-''e-2>a-' 'f2>b-''e-8.>a-''f>b-16^4'
567: 'f2>b-''e-2>a-' 'e-2>a-''f2>a-' 'f+1c''f1>f'
568:
569: |:
570: o5u80
571: 'c+1>>b-f''c+1>>af' 'c+1>>a-f''e-1>>b-e-'
572: 'e-2>>b-e-''c2>>a-c' 'c+2>>a-c+''>b-2>f>b-'
573: 'f+1>>b-c''f1>c>f'
574:
575: 'c+1>>b-f''c+1>>af' 'c+1>>a-f''e-1>>ge-'
576: 'e-2>>a-e-''c2>>a-c' 'c+2>>fc+''>b-2>f>b-'
577: 'c2>>f+c''f2>f>f' 'c+1>b->c+>b-'
578:
579: o5u62
580: |:
581: 'c+2>b-c+''c2>a-c' '>a-2f>a-''c+2>b-c+'
582: 'f+2>b-f+''e-2>a-e-' 'c+1>fc+'
583: 'c+2>b-c+''c2>a-c' '>a-2f>a-''c+2>b-c+'
584: 'f+2>b-f+''e-2>ae-' 'c+1>fc+'
585: :| | o4u60q8'b-1fc+>b-'
586: :|
587: 
588: o5u68@q1
589: 'f2c+>f''>a-2e->a-' 'e-2>a-e-''c+2>fc+'
590: 'b-2f+>b-''e-2>a-e-' 'a-2f>a-''f2c+>f'
591: 'f+2>b-f+''e-2>a-e-' 'e-2>a-e-''c+2>fc+'
592: 'f+2c+>f+''f2c>c' 'b-1f>b-'
593:
594: i0,2@43@v72u125
595: o3q8'f+1>f+1''a-1>a-'
596: 'c+1>c+''c+>c+'>'a->a-'<'c>c''f>f'
597: 'f+1>f+'L8'f>f2^''f>f''e->e-''f>f'
598: 'b->b-2^''b->b-''a>a''b->b-' <'c+>c+4.'>'b->b-4.''e>e4'
599: @v80'e-.>e-''f.>f''f+>f+''a-2>a-'
600: 'f+4.>f+''c+>c+''f+.>f+'>'b-.>b-'<'e->e-'
601: @g12'f>f1^16'&@b0,-8192'f>f8'r16r2.@b0r2
602:
603: i1@49o5@v80u62r2
604: |:4
605: 'd2>bd''c+2>ac+' '>a2f+>a''d2>bd'
606: 'g2>bg''e2>ae' 'd1>f+d'
607: 'd2>bd''c+2>ac+' '>a2f+>a''d2>bd'
608: 'g2>bg''e2>a+e' 'd1>f+d'
609: :|
610:
611: /=====================================================
612: (t16)
613: n8i8,2@3@v100u100p3r                        / piano
614: r2
615: r1r1 r1r1 r1r1 r1r2
616: L8rrrr4.
617: u90o4L8q7'b-4<fb-''b-4<fb-'
618: q6'b-.<e-a-''b-.<e-a-'q4'b-<e-a-'
619: rq3'a-<c+f'q8'a-4<c+f'q7'b-.f<c+''b-.f<c+'q4'b-f<c+'
620: q7'b-4<c+f''b-4<c+f' q6'b-.a-<e-''b-.a-<e-'q4'b-a-<e-'
621: rq3'a-f<c+'q8'a-4f<c+'q7'a-.f<e-''a-.f<f'q4'a-f<f'
622:
623: u90o5q7'b-4f>b-''b-4f>b-'
624: q6'a-.e->b-''a-.e->b-'q4'a-e->b-'
625: rq3'fc+>a-'q8'f4c+>a-'q7'f.>b-f''f.>b-f'q4'f>b-f'
626: u-10q7'f+4e-c>f+''f+4e-c>f+'
627: q6'f+.e-c>f+''f+.e-c>f+'q4'f+e-c>f+'r1u
628:
629: |:
630: u80o5q7'c+1>b-f>b-''c+1>af>a'
631: 'c+1>a-f>a-''e-1>b-e->b-'
632: 'c2>f+e->f+''c2>a-e->a-''c+2>f>b-f''>b-2c+>b-f'
633: 'c1>f+c>b-''f+8.c>af+''fc>af2.^16'
634:
635: 'c+1>b-f>b-''c+1>af>a'
636: 'c+1>a-f>a-''e-1>b-e->b-'
637: 'e-2>a-e->a-''c2>a-c>a-'
638: q5u+20'f8.>f''e-8.>f''f>f'q7u-10'c+2>fc+>f'u
639: u+15'c2>f+c''f2c>f'u-10'c.>fc'u'>b-f>b-2.^16'
640:
641: |:
642: o5u84q7'f+4>b-f+''f+4>b-f+''f.>a-f''f>a-f16^4'
643: q5r'e->f+e-'r'e->f+e-'q6'c+.>fc+'q5'c+.>fc+'q5'c+>fc+'
644: q7'f+4c+>f+''f+4c+>f+'q6'e-.c>e-''e-.c>e-'q5'e-c>e-'
645: rq4'c+>a-c+''c+>a-c+'rq7'c+.>a-c+'q6'f.c+>f'q5'fc+>f'
646:    q7'f+4>b-f+''f+4>b-f+''f.>a-f''f>a-f16^4'
647: q5r'e->f+e-'r'e->f+e-'q6'c+.>fc+'q5'c+.>fc+'q5'c+>fc+'
648: q7'f+4c+>f+''f+4c+>f+'q6'e-.>ae-''e-.>ae-'q5'e->ae-'|
649: rq4'c+>fc+''c+>fc+'rq7'c+.>fc+'q6'f.>b-f'q5'c+>fc+'
650: :| r1 |r1
651: :|
652:
653: r1r1 r1r1 r1r1 r1r1
654: r1r1 r1r1 r1r1
655: r2.i5@125@g12@y1,$63,76u45r8o4@b0,3415'b>a>eg1^8'
656: r1i8@3@y1,$63,64@b0r1 r1r1r1
657:
658: |:4
659: o5u84q7'f+4b>f+''f+4>bf+''e.>ae''e>ae16^4'
660: q5r'e>ge'r'e>ge'q6'd.>f+d'q5'd.>f+d'q5'd>f+d'
661: q7'g4d>g''g4d>g'q6'e.c+>e''e.c+>e'q5'ec+>e'
662: rq4'd>ad''d>ad'rq7'd.>ad'q6'f+.d>f+'q5'f+d>f+'
663: q7'f+4b>f+''f+4>bf+''e.>ae''e>ae16^4'
664: q5r'e>ge'r'e>ge'q6'd.>f+d'q5'd.>f+d'q5'd>f+d'
665: q7'g4d>g''g4d>g'q6'e.>b-e''e.>b-e'q5'e>b-e'
666: rq4'd>f+d''d>f+d'rq7'd.>f+d'q6'f+.>bf+'q5'd>f+d'
667: :|
668:
669: /=====================================================
670: (t17)
671: n9i5,2@81@v50u100@p54r                      / rhythm sub
672: @y1,$21,82
673: y99,1 y98,$20       / (y6,n = @y1,$20,n)
674: r2
675: r1r1 r1r1 r1r1 r1r2
676: L8rrrr4.
677: o3u100L16@q10
```

▶なんか最近シャープの製品の値段が気になる。新ザウルス，68,000円。プロジェクタ，680,000円。ポケットMD，68,000円。たんに「6」と「8」に反応しているだけか……。
新谷　貴幸(20)埼玉県

```
678: |:8y6,70b-b-b-ry6,75b-b-b-r
679: y6,80b-y6,85b-b-ry6,90b-y6,100b-y6,80b-r:|
680:
681: |:
682: |:32y6,70b-b-b-r y6,75b-b-b-r
683: b-y6,80b-b-r b-y6,90b-y6,85b-r:|
684: |r1:|
685:
686: |:8y6,70b-b-b-r y6,75b-b-b-r
687: b-y6,80b-b-r b-y6,90b-y6,85b-r:|
688:
689: r1r1 r1r1 r1r1 r1r1
690: r1r1 r1r1r1
691:
692: |:32y6,70bbbr y6,75bbbr
693: by6,80bbr by6,90by6,85br:|
694:
695: /==================================================
696: (t18)
697: n12i8,2@62@v70u102p1r                        / brass (L)
698: @y1,$20,74@y1,$21,68
699: (t19)
700: n13i0,2@62@v68u110p2r                        / brass (R)
701: @y1,$20,78@y1,$21,72
702:
703: (t18,19)
704: r2 @y1,$63,48
705: r1r1 r1r1 r1r1r1r2
706: L8rrrr4.
707: r1r1 r1r1 r1r1 r1r1
708:
709: |:r*3072
710: |:o5L8
711: r1rq4fre-@q1c+.e-.q3f
712: r1rq4fre-q6_6'f.c+'q7'f+.e-''a-f'~6
713: r2.rq3u+4f+urq4fre-@q1c+.e-.q3f
714: r1rL16@q1u-5'c+<c+'>'b-<b-''a-<a-''b-<b-'r<|
715: 'c+8.<c+''e-<e-'u-10'f<f'u+10
716: u-10'c+<c+'>u-10'a-<a-'<'c+<c+'>'b-<b-'L8<u
717: :|
718: L16o4'b-<b-'r'b-<b-''a-<a-''a<a'L8'b-<b-'r
719: |r1:| r*4032
720:
721: |:
722: o5L8
723: |:4rq3f+rq4e|@q1d.e.q3d:|q6_6'f+.d'q7'g.e''af+'~6
724: |:3rq3f+rq4e @q1d.e.q3d:|
725: rL16@q1u-5'd<d'>'b<b''a<a''b<b'r<
726: 'd8.<d''e<e'u-10'f+<f+'u+10
727: u-10'd<d'>u-10'a<a'<'d<d'>'b<b'L8<u
728: |:4rq3f+rq4e|@q1d.e.q3d:|q6_6'f+.d'q7'g.e''af+'~6
729: |:3rq3f+rq4e @q1d.e.q3d:|
730: rL16@q1u-5'd<d'>'b<b''a<a''b<b'r<
731: o4'b<b'r'b<b''a<a''b-<b-'L8'b<b'r
732: :|
733:
734: /==================================================
735: (t20)
736: n14i0,2@41@v50u100p3r                        / violin
737: @y1,$20,74@y1,$21,68
738: r2
739: r*17136
740:
741: @v62u106
742: @m20o5L4q8@y1,$63,69f1^e-^f
743: @y1,$63,76@m25c+1c1
744: >@y1,$63,66b-1<L8c2^@y1,$63,46@y1,$66,82q6cc+e-
745: @y1,$63,73@y1,$66,64q8f2^@y1,$63,46@y1,$66,82q6fef
746: @y1,$63,67@y1,$66,67q8<L4c+>b-fe
747: @y1,$63,65@y1,$66,64@q2L8u-3e-.u-3f.f+ f.f+.a-
748: f+.a-.b- <c.u-5c+.e- u+3f1r1r1
749:
750:
751: (p)
```

リストフ　トゥインクルトゥインクル用カウンタ表示

```
 1:00006540 00000000   2:00006540 00000000   3:00006540 00000000   4:00006540 00000000
 5:00006540 00000000   6:00004D40 00000000   7:000006C0 00000000   8:00004D40 00000000
 9:000006C0 00000000  10:00006540 00000000  11:00006540 00000000  12:00006540 00000000
13:00006540 00000000  14:00006540 00000000  15:00006540 00000000  16:00006540 00000000
17:00006540 00000000  18:00006540 00000000  19:00006540 00000000  20:00004D40 00000000
```

(善)の「勝負はこれからだ」

ある日ものすごい集団がいるのを知った。なんとファルコムの「ソーサリアン」をX68000上で動かしてしまうパッチプログラムを開発してしまったというのだ。CPUも違う，ソースも資料提供もされない，参考資料ゼロの状態で独自に解析を行い移植してしまったということらしい。もちろん形態はパッチプログラムなのでプレイするには製品版のソーサリアン(PC-8801，PC-9801，X1turbo版のいずれでもよい)が必要で，しかもそのパッチ当てプログラムはPC-9801専用なのでPC-9801も必要だ。なにからなにまでまったくそっくりにできていて，さらにゲーム途中セーブ機能やバックログ機能などのX68000オリジナルの機能も満載で，とても個人で開発したとは思えない完成度。

ところでパッチ当てにPC-9801が必要ってことはPC-9801を持っていないとダメなんだから，だったらPC-9801用のソーサリアンを買って遊べばいいじゃないかということになってくる。が，この一見無意味な行動に私は言葉では表すことのできない，冒険心とロマンを感じた。「そこに山があるから登る」登山家と同じように，「そこにソーサリアンがあったからX68000に移植した」のではないだろうか。

しかしこんなユーザーがいる限りX68000はパソコンを超えたひとつの趣味表現の道具として生き続けそうな気がする。「パソコンなに使ってるの？」と聞かれて本当のこというと異教徒扱いを受けそうで怖いってことで「んーまぁ一般的なヤツを使ってるよ」などとついうそぶいてしまい，罪悪感に苦しんでいるX68000ユーザーも多いことと思う。電波新聞社のベーマガを開いても，X68000の新作コーナーはもはや葬式ごっこをやられている，いじめられっ子の心境で，妙にジャガーに親近感を覚えてしまったりする。でも，もういいのだ。みんなX68000ユーザーであることを誇りに思うのだー。

日本語ラップの話

最近CMでよく耳にする日本語ラップミュージック。私はどれもこれもチョーダサく感じてしまっているのだが，どうだろうか。

まずペヤングソース焼そばのシノスケ。「安くてグーうまくてナイス，食べれば，食べるとき……以下略」なんと早口言葉，挙句の果てに動詞の活用，まったくラップをバカにしてる。これと同種にいくえちゃんのムシューダがある。

最近台頭してきた「いいじゃん」を多用するあの変てこなヤツ，資生堂「ティセラ」。確かに独創的ではあるが聞いてて全然気持ちよくない。あの踊りを踊っているバンダナ巻いた頭をカクカクしているガキどもにはぜひ「SOUL II SOUL Volume3 just right」でも聞いてソウルミュージックのなんたるかを学んでほしいところだ。

もともとラップは黒人キリスト教徒のミサで歌われる前衛的な賛美歌が起源で，メロディなしのリズムに乗せて自分たちの魂の叫びを歌ったのが始まりだ。独特の黒人アクセント，韻を踏むように(これをrhyme/ライムというのだ)巧みに構成された詞，これらが独特の音楽性を醸し出し，広まっていったのだ。いまの日本語ラップは明らかに猿マネ以下で，その多くはいちばん基本の押韻さえしない。単にリズムに早口の旋律なしのおしゃべりが乗っているだけ。いま一度この日本語ラップ，見直してほしいものだ。日本語は英語ほど抑揚がないからラップには合わないんじゃないかと気づき始めているミュージシャンもいるにはいるみたいだが……。

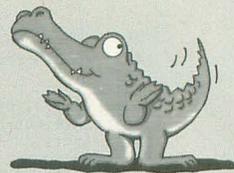
さて，私はここで日本語ラップ撲滅委員会を設立する。自分が耳にした日本語ラップ曲のダサさを具体的に明記したハガキを送ってくれた人はもれなく委員になれるぞ。

横浜駅毒ガス事件の話

4月21日17時30分横浜駅で毒ガス事件が発生した。そのときなんと私は現場近くにいたのだ。

私はその日発売された餓狼伝説3を買うために横浜駅からヨドバシカメラに向けて小走りしていた。事前に電話でヨドバシに在庫を確認したが残り少ないので急いでくださいといわれて焦っていたのだ。と，そのとき後ろから男の大声が。後ろを振り返るとなんと警察官の大群が私に向かって近づいてくるではないか。なにごと，と思っているうちに小走りの私をあっという間に追い抜き走り去って行った。また毒ガス事件でもあったか？　と思いつつも私は餓狼3を求めて小走りを続けた。そして後ろにまた駆け足の音が聞こえてきたので不思議に思い後ろを向くと……それは，さっきの警官集団がなにしに行ったのかを確かめたい野次馬集団が，私の小走りにつられ一斉に走り出した音だった。かくして私は野次馬集団の先導者になってしまったのだった。　（善）

▶アキフ・ピリンチ作「猫たちの聖夜」(早川書房)は，大変面白かったです。被害者も探偵も犯人(？)も，すべて猫というサイコサスペンス……という以上の物語でした。
川井　健司(33)埼玉県

(善)のゲームミュージックでバビンチョ

西川善司

●交響曲イース'95
CD:KICA-1160　3,000円(税込み)
キングレコード　5/24発売

今回発売された「交響曲イース'95」はイース1のオーケストラアレンジバージョン。ストーリーを追うような形で編曲がなされ全5楽章構成になっている。なぜいまさらイース1なのかはよくわからないが，いま思い返してみると，メロディが自然と頭に蘇ってくる曲というのは全イースシリーズを通してやはり「1」のものが断然多い。しかし，名曲である半面，それだけ原曲のFM音源サウンドの印象が強いという意味もあるわけで，アレンジする側にとっては非常に難しい題材でもあるわけだ。

アレンジャーに関する資料は手元になにもなかったが，とにかく原曲のフレーズのイメージを生かした各楽器へのパート振りがお見事。編曲も素晴らしい。第1楽章はフィーナのテーマをロンド形式で綴ったもので，途中から草原のテーマにつながる組曲形態。音楽的に非常に優れている。

いままでイースネタのアレンジバージョンCDはずいぶんと出ているが，これが最高峰なのは間違いない。しかしこれって打ち込みなの？

・おすすめ度　10

●愛・超兄貴/岩崎琢
CD:TYCY-5437　2,800円(税込み)
東芝EMI　5/17発売

一部に熱狂的ファンを獲得しているこのゲーム，さっそくオリジナルサウンドトラックが発売になった。しかし今度の音楽担当は葉山ではなく別の人。熱狂的な兄貴ファンたちはこれをどう受け止めているのかは私の想像の及ぶところではないが，私のファーストインプレッションは「面白い!」のひと言に尽きる。GMを聞いてゲームをやりたくなったのはバーニングフォース以来だが，ちょっとあのときとは違った意味でゲームが気になってきた。たとえばTR1の「上海パワースラム」。YMOを彷彿させるアジア色の濃いメロディにパーカッション的に被さるのはもうお決まり，兄貴のかけ声。なんとも異様……なのにカッコイイ。その他，壮大な弦楽器による交響詩に合わせて叫ぶオペラ調コーラス隊，レトロなムードミュージックを奏でる蓄音機の前でラブシーンをおっぱじめるフランス人男女など，常人の理解を超えた楽器(?)が世紀末的な音楽を展開する。まさに音楽のびっくり箱。音楽でこれだけ笑いをとる岩崎琢，葉山をすでに超えている?

・おすすめ度　9

●ツインビーPARADISE 2 Vol.6
CD:KICA-7660　2,800円(税込み)
キングレコード　5/24発売

文化放送系で放送されていたツインビーのラジオドラマを収録したお馴染みのツイパラ2がついに今回のVol.6をもって完結する。巨大隕石地球接近の危機(ツイパラ1最終話)から逃れた人類はあれから1年も経たずに再び地球滅亡の危機に瀕してしまう。ワルモン博士の作ったブラックツインビーのパイロットである人工生命体の姉弟が再び登場。ツインビーチームと彼ら姉弟の対決は? ワルモン博士の製作した最終兵器ブラックホール発生機の威力は？　前作最終話で正義の味方になってしまったワルモンは今度はどうやって涙を誘うのか。感動の大団円に乞うご期待といったところ。おまけの声優陣によるサイコロトーク，そしてテーマソング「瞳はVenus」のAnotherバージョンも収録。

・おすすめ度　7

●ナムコゲームサウンドエクスプレス
Vol.18　エアーコンバット22
CD:VICL-15040　1,500円(税込み)
ビクターエンタテインメント　5/24発売

あの本格ドッグファイト3Dシューティングがナムコ最新ハードシステムスーパー22で蘇った。前作のシステムよりも格段にハード性能が向上したおかげでその映像インパクトはもはや「リアル」という言葉の意味を超えている。

前作はいかにもキーボディストの曲といった感じのシンセサウンド中心の爽やか系の曲が多かった。しかしやはりドッグファイトにはHM/HR(ヘビメタ/ハードロック)が似合うということで，今回はギターサウンド中心の曲が多くなっている。とはいえ，前作で目立っていたあのフォ～っとした爽やかなシンセパッドが要所要所にヘビーなギターを包みこむように挿入されてくるので，この辺が「前作からのイメージの継承」という印象をリスナーに与えてくれる。

このゲームはビデオのほうの発売も期待したいねぇ。

・おすすめ度　7

●バーチャファイター
「ザ・エターナルバトル」
VHS:TYVY-5006　4,800円(税込み)
東芝EMI　発売中

1995年2月12日，横浜ジョイポリスで行われたバーチャファイターの統一チャンピオン決定戦の模様を収めたビデオ。ゲームの攻略/紹介は一切なし。純粋にこの大会の模様をレポートしたかなり玄人向けの内容だ。この大会はセガサターン版バーチャファイター，アーケード版バーチャファイター，そして最新のバーチャファイター2の3タイトルすべてをプレイしてその腕前を競う大会。トーナメントは一般の格闘技のように男子の部，女子の部と性別分けされている，というこだわりよう。SS版の対戦は立ち&パッドという過酷な条件下でのプレイを強いられていたようだが，アレにはどんな意味が？

・おすすめ度　7

交響曲イース'95

エアーコンバット22

［特別企画］

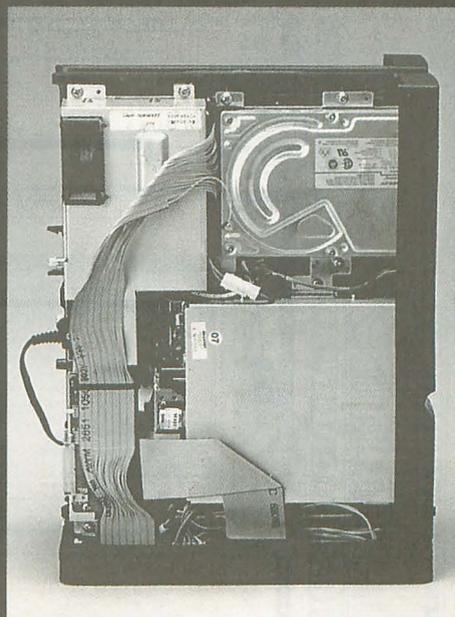
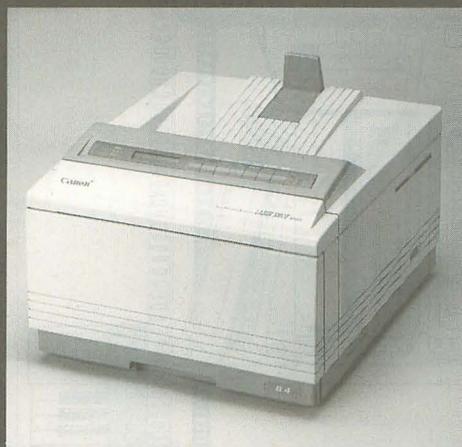
X68000周辺機器パワーアップ計画

PERSONAL WORKSTATION

X68000 XVI SERIES

POWER TO MAKE YOUR DREAM COME TRUE

パソコンでできることが多様化し，それに合わせて必要なものもかなり増えてきている。本体のみでひととおりのことを楽しめるX68000でも，その状況は変わらない。やろうとすることが高度化するにつれ，メインメモリ2Mバイトでは動かないものが多くなり，フロッピーベースでの作業はすでに過去のものといえる。しかも，ここ最近状況がさらに加速しているのだ。しかし，メーカーのサポートをあてにできない現在において，新たに周辺機器の購入に踏み切ることは非常に勇気がいる。一度，X68000が投資するに値するマシンかどうか，じっくり考えてみるといいだろう。その結果，X68000をより楽しみたいという欲求があるのであれば，必要なものを揃えてしまうべきだ。
いまが決断のとき。周辺機器の充実を図ろうではないか。

CONTENTS

目指せ周辺機器貧乏

Hamazaki Masaya 浜崎 正哉

夢を超えたX68000

X68000は，本体のみでもひととおりのことができてしまうため，多少のメモリと贅沢品であったハードディスクがあれば，快適なパソコンライフを送ることができました。しかし，ここにきてソフトウェアや扱うデータの肥大化にともない，本体のみでパソコンを楽しむことは難しくなってきつつあります。ゲームをするにもメモリやCPUパワーが必要なものがあり，なかなか貧乏人にはつらい状況です。

ひと昔前には，メインメモリが2Mバイトもあれば困らないような幸せな世界だったのに……。あっという間にフルメモリ実装，ギガバイトハードディスクでも足りない世界です。時の流れってやつは……。

これまでの周辺機器事情

イソノ界（サザ○さんのような，主に漫画世界で時間の流れがリサイクルしている状態を指す）にとりこまれたような時間が停滞している現在のX68000。世の中は日々進歩し続け，さまざまな周辺機器が登場してきました。もちろんX68000を正式サポートしているものは，数少ないものです。

といってもそういったおいしそうな周辺機器をパワーユーザーが見逃すはずもありません。ハード的に接続が可能であれば，対応ソフトを自作したり，ハードまでも自作するなどという涙ぐましいまでの努力が行われてきました。

ただし，そこまでこだわって実際に使いこなせる人は，いくらX68000ユーザーといえども限られます。普通にコンピュータを楽しんでいる人は，正式に対応がアナウンスされているサードパーティの製品を使うしかなく，高機能な周辺機器を使う他機種ユーザーを横目でうらやましがるしかなかったのです。

現在の周辺機器事情

こうしたユーザーと一部メーカーの努力のかいあって，現在では正式サポートのない製品でも，使うためのノウハウが集まってきました。わずかながら，アプリケーションも各種周辺機器に対応してきています。それなりに制限がありますが，以前の状況に比べれば格段に進歩したといえるでしょう。巷に氾濫している，他機種用に安価に発売される高機能な周辺機器が，X68000でも使えるのですからね。

確かにグラフィックを扱うのであれば，実質MATIERでサポートしているものが標準ですし，CD-ROMもCDDEV.SYSで使えるものを中心に製品選びをする必要があります。MIDIに関しては，ハード的な問題はありませんが，シーケンサソフトのラインナップの貧弱さは隠せません。

しかし，メーカーサポートの貧弱さはいまさらですから，文句はいわないことにしましょう。使える周辺機器があるだけでも幸せってものですよ。快適なパソコンライフを過ごせるのであれば，多少のリスクくらい覚悟して受け入れましょう。

周辺機器充実のススメ

ということで，まず，周辺機器の筆頭に挙げられるのがメインメモリ，大容量デバイスです。ともにあればあるほど操作環境を向上させてくれます。

あとは，各自がやりたいことに合わせて周辺機器を購入していきましょう。処理速度に不満があるなら，最近にわかに活気づいている，アクセラレータに手を出すのもいいかもしれません。不思議なもので，メモリを増設しただけでも，なんだかパソコンを使ってやろうという気分にさせてくれるものです。近ごろマンネリ気味でパソコンがつまらない人は，ちょっとだけ自分のパソコンに投資してみるといいかもね。

ただし，現在のX68000の世界は，目的のものがなんでも揃っているようなパラダイスではありません。過剰な期待を抱くと冷たい現実の前にくじけてしまいます。注意しましょう。

結論

・メインメモリは12Mバイト
・必要かもしれない，そう思ったらすぐに買え
・衝動買い大いに結構
・後悔なんかしてはいけない。ひたすら使うことを考えろ
・最後に愛が勝つ
・そして，皆貧乏になろう！

まあ，半分冗談，半分本気ですが，結局，自分のやりたいことをよく考えたうえで，なにが必要なのか見極めて購入するしかありません。

使うのは自分です。必要と判断したら思い切って買うべきです。無駄な投資にならないか，なんて躊躇している暇があったら，めいっぱい楽しんで元を取ることを積極的に考えるようにしましょう。

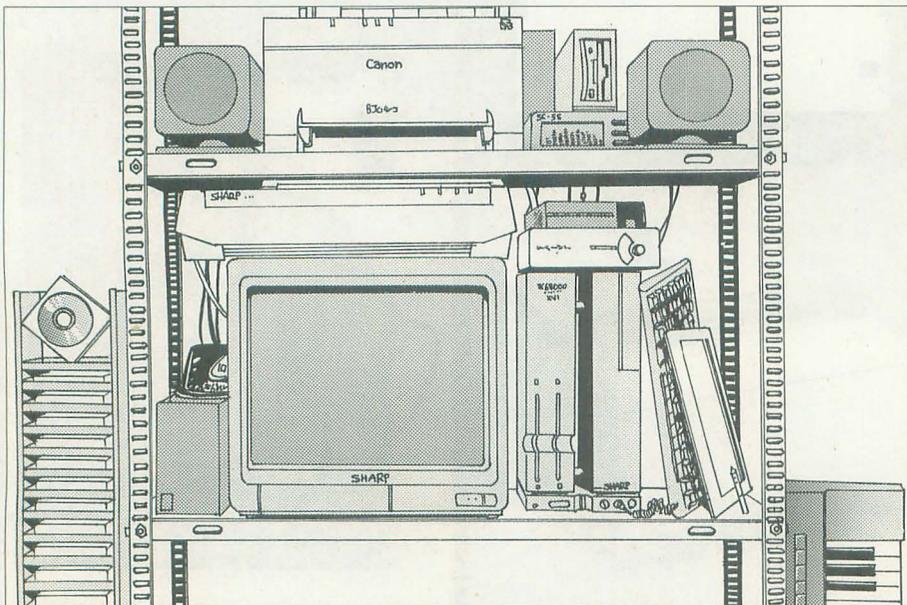



illustration:T.Takahashi

メモリ増設あれこれ

Sakamaki Katsumi 坂巻 克巳

● 増大する平均メインメモリ ●

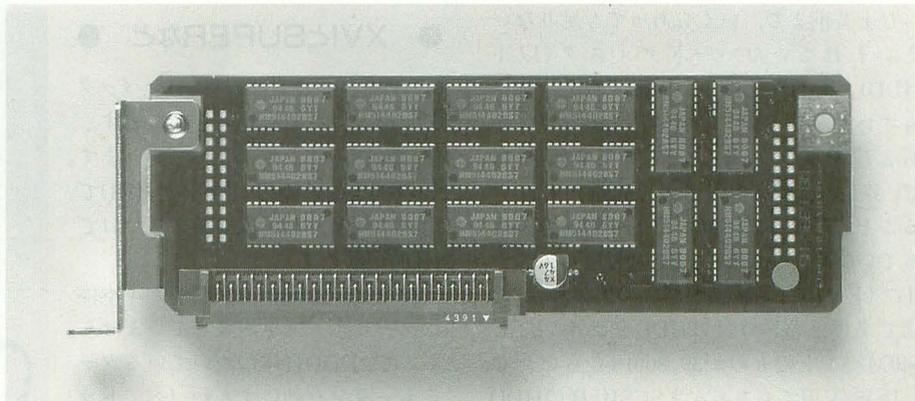
今月号のアンケート分析大会によると，平均メインメモリが6Mバイトと昨年に比べ，ずいぶんと上がりました。12Mバイトフルメモリと，2Mバイトがほぼ一緒という興味深い結果です。

これは，ここ最近サードパーティの製品，特にSIMMメモリボードの登場により，比較的安価にメモリを増設できるようになったことが挙げられます。下手に2Mバイトごとにちまちまと増設するよりも，4Mバイトもしくは8Mバイトを一気に増設したほうが安上がりなのですから当然ですね。メモリは多いぶんには困りませんから。

● メモリはどれだけ必要なのか ●

さて，多いぶんには困らないメモリですが，用途によってどれだけのメモリが必要なのでしょうか。簡単にみていきます。

コマンドシェルでプログラミングを主体とするなら2Mバイトでも十分用が足りるでしょう。アセンブラレベルであれば，SLASHを使った開発もできてしまいます。ただし，同じプログラミングでもC言語，特にGCCを使用するとなると4Mバイトはほしくなります。さらにSX-WINDOWではプログラミングどころか使うだけでも，最低ラインが8Mバイトとなります(場合によっては12Mバイトでも足りないくらい)。

グラフィックや音楽に関しても，年々扱うデータ量が増大しています。MATIERでも実質4Mバイトないとうまみがありませんし，Z-MUSICでPCMデータを本格的に扱うとなれば，2Mバイトではとうてい足りなくなります。また，増大するデータのほかにも，非力なCPUパワーをカバーするために，メモリが湯水のごとく使われることもあります。あまり安易にメモリを消費するのも考えものですけどね。そして，快適な環境を目指すのであれば，RAMディスクやディスクキャッシュも確保したくなります。なければないですませられますが，とおっても快適ですよ。

目指せフルメモリ

● 増設メモリ ●

増設メモリボードには本体内蔵のスロットに差すタイプと，拡張スロットに差すタイプがあります。とりあえず，メモリボードをスロットに差して，SWITCH.Xでメモリ容量を設定すれば，無事増設したメモリを使えるようになります。で，問題なのは，高速タイプ(X68000 XVI以降)のマシンで，拡張スロット用のメモリを使った場合です。このとき，そのメモリをアクセスするたびに必ずウェイトがかかります。これは，従来機との互換性のために，拡張スロット上のメモリは，必ず10MHzでアクセスするようにハード的な処理がされているためです。ですから，いくら速いSIMMメモリを使用しても，メモリが拡張スロットに登載されている限り10MHzの壁に阻まれ，思うように性能が発揮されません。

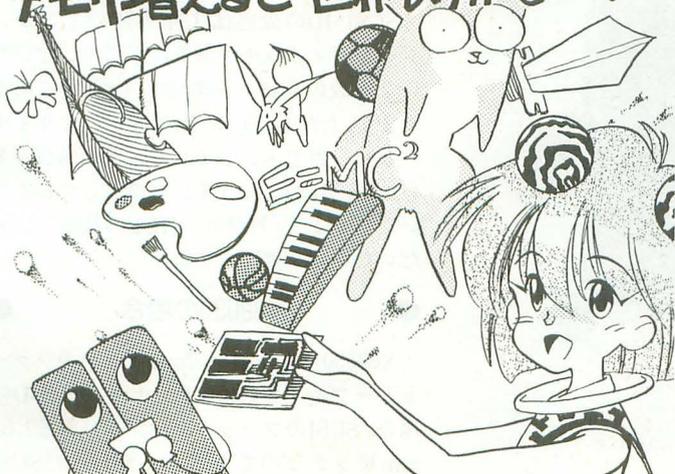
● メモリチェック ●

メモリを増設したはいいけど，心配なのはその増設メモリがちゃんと動いているかどうかです。特に安いSIMMメモリを登載した場合，正常に動作するかどうか，結構不安がありますからね。

いちばん簡単なメモリチェック方法は，片っ端からメモリ内容を読み出して，バスエラーが発生しないかどうか調べるだけです。

次にデータ化けのチェックですが，メモリに特定の値(10001000_Bがいいらしい)を書き込んでおき，数時間経過したあとにメモリ内容を調べればOKです。チェックプログラムをSRAM上に置くようにすれば，問題ないでしょう。

どちらもアセンブラでプログラムを組むことができれば，特に問題なく簡単にチェックプログラムを作成できるでしょう。

プログラミングの不得意な人には，SX-WINDOWでフリーエリアぎりぎりまで，なんらかのアプリケーションを動かし続けるという方法もあります。あまり確実な方法ではありませんが，あまり頻繁にハングアップするようであれば，メモリを疑ったほうがいいでしょう。

大容量ハードディスク導入手引き

Taki Yasushi 瀧 康史

ひと昔前まで，いくらあっても足りないといわれていたハードディスク(以下HDD)。皆さんは足りていますか？ 価格もずいぶんと下がり，540Mバイトクラスなら2万円切っている(ただし秋葉原価格)ので，多くの人が大容量の恩恵に預かっていることでしょう。730Mバイトでも3万円切っています。もちろん，DOS/V用ベアドライブ(ドライブユニットだけの裸のもの)の話ですけどね。外付けHDDドライブだと540Mバイトのものが39,800円ぐらいかな。

DOS/V用でもちゃんとSCSI仕様のHDDならX68000には使えます（ちなみにIDEのものは使用できませんので購入しないように）。3.5インチのSCSIベアドライブはX68000でも共通なのです。この規格が決まったのはなにも最近の話ではなく，SCSI-HDDが出回り始めた当初からずっと同じなのです。

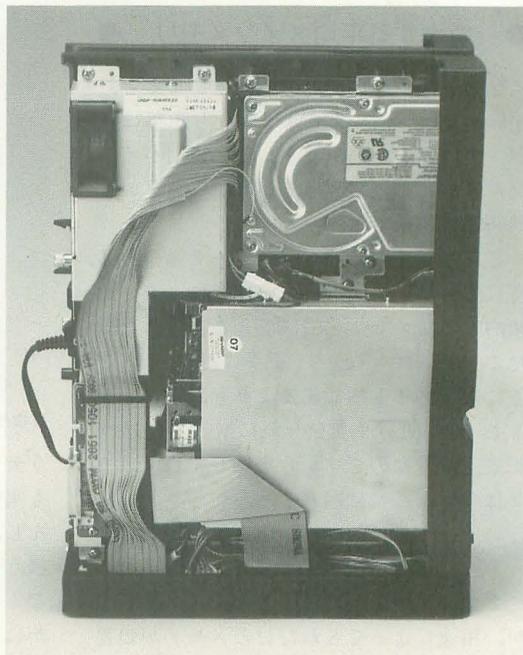
要するに，SUPER-HD，XVI-HDの中に入っているHDDを，最新のDOS/V用HDDに入れ替えることができます。もちろん，昔使っていたような，ITECのTX80などにも入れ替え可能です。

また，HDDさえ買ってくれば，SUPER，XVIユーザーでHDDを増設していない方もDOS/V用大容量HDDの搭載が可能です。

● XVIとSUPERなど ●

HDモデルならば，HDDをバックアップし，左のタワーを開けます。左のタワーは黒ネジ3つで開きます。上のほうにある黒いブツがHDDです。これを外し，取り換えるだけでHDDが動きます。

ただし，X68000のHDモデル本体はターミネータを内蔵していません。そこで，HDD自体についているターミネータを有効にするか（ベアドライブのマニュアルを参考にしてください），SCSIポートの手前にある一連のソケットに，330Ωと220Ωの抵抗をつければOKです。ただし，抵抗をつけるよりも，最近安くなったアクティブターミネータを使ったほうが信号ラインは安定します。

HDなしモデルなら，XVI用HDDアングル(流通コード007 200 0177：500円)と，XVI用HDD電源ケーブル(流通コード0007 512 0258：1,400円)，XVI用HDD接続ケーブル(流通コード007 512 0257：3,000円)を購入してください。HDD接続ケーブルは，HDDなしモデルの場合，代わりの短いケーブルが本体に付属していますから，それと取り換えます。

安価で大容量のハードディスクがあなたの手に……

720MバイトタイプのQuantum製ドライブLIGHTNING 730Sと必要な部品類

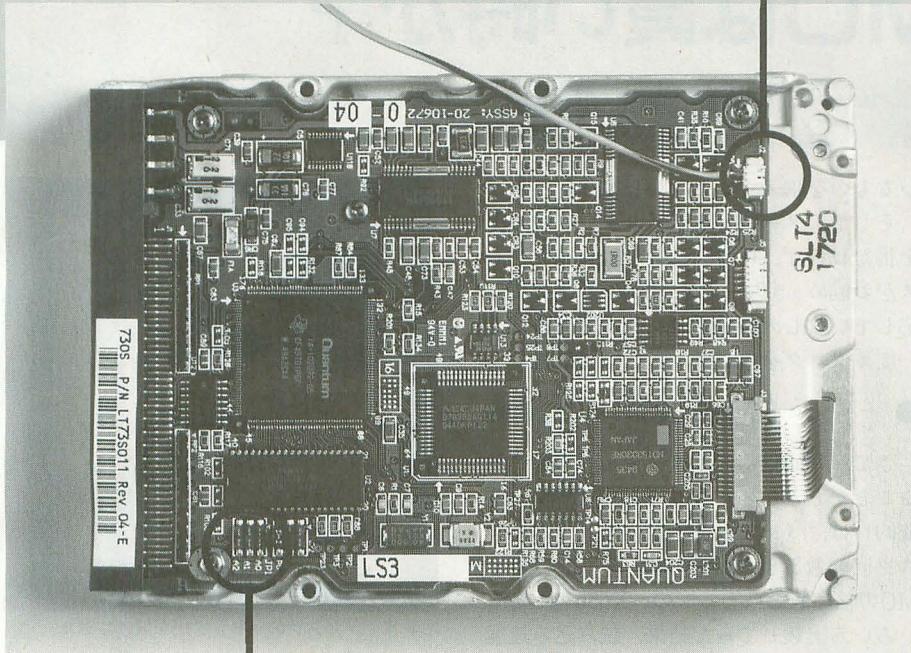
HDDの取りつけはたったこれだけで終わります。この状態で電源を入れ，FORMAT.Xを実行すると，HDDが認識されているはずです。ただし，このままでは，HD-BUSYランプはつきません。HD-BUSYをつけるためには，フラットケーブルの36ピンを切って，HDDの基板面にあるLEDのカソード側に接続します。間違えても壊れないはずなので，アクセスLEDの足の適当なほうに接触させてみてください。これで，本体右側のHD-BUSYが点灯するようになります。

SCSI-IDの設定はHDD本体のどこかにジャンパスイッチがあるはずです。だいたい2進数の3ビット構成で，0～7まで決まります。ただし，SCSI-IDは固定になります。

なお，固定ネジはインチネジの場合が多いので，HDDに合うネジを購入しておくといいでしょう。外付けのHDDの接続方法もだいたい同じ要領ですね。

● X68030の場合 ●

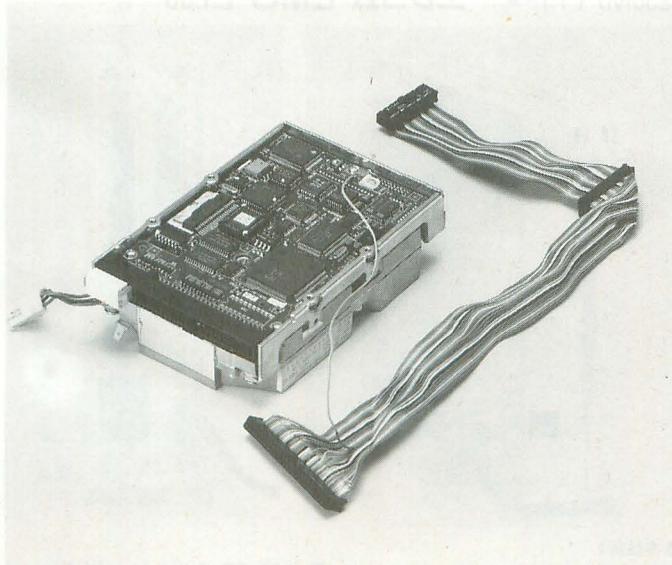
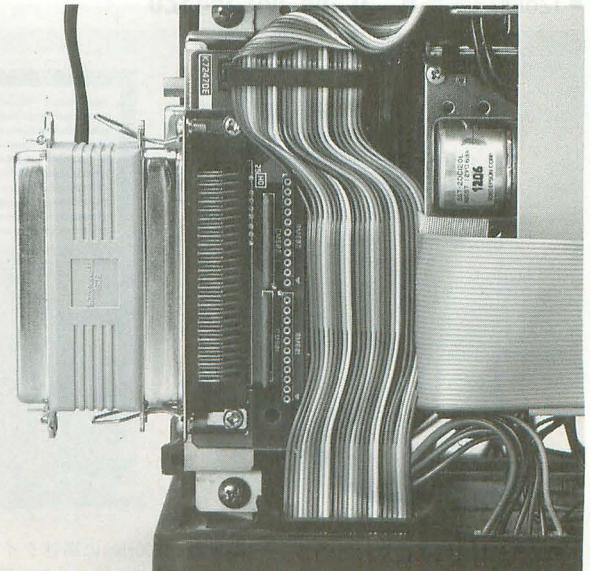
X68030は本体側が，2mmピッチのフラットケーブルです。3.5インチSCSI HDDの場合，SCSIのフラットケーブルは普通2.54mmピッチなのでちょっと変わったSCSI

アクセスランプの信号線を接続。別に接続しなくても動作はするが，やはり接続しておいたほうが気分はいい

IDはジャンパスイッチで設定。とりあえず内蔵ハードディスクということで，ID＝0にしてある。本体側の内蔵用ID設定スイッチは今回の改造だけでは使えない

ケーブルを作らなければなりません。というのも，内蔵SCSIケーブルは2.5インチHDD用だからです(流通コード007 512 0302：40,000円)。2.5インチ用アングルは，もともと本体側についているので，ケーブルだけ買ってくれば，2.5インチSCSI-HDDを接続することができます(ただし，2.5インチドライブは大半がIDEなので注意)。

とはいっても，2.5インチSCSI-HDDは高価です。3.5インチを作る場合，ピッチを越えた鬼のような作業をして接続しなければいけませんが，私は，ケーブルの50ピン側（2.5インチ用）をバラして，2.54mmピッチの50ピンに，しかも3.5インチSCSI-HDDをそのままつけられるように改造しています(ケーブルの配線は表)。なお，3.5インチHDDを利用する場合，アングルがそのままではつかえないので，ちょうどよい位置に穴をあけるなり，500円のXVI用HDDアングルを買ってきてください。また，電源ケーブルは1,400円のXVI用のものを買ってくる必要があります。

表　CZ500シリーズ内蔵HDD用SCSIケーブル

CZ500側ケーブル番号	SCSI低密度型ノンシールドコネクタ
1.N.C.	IDに接続
2.N.C.	IDに接続
3.N.C.	IDに接続
4.HD LED	HDD LEDのカソードへ
5.VCCI(+5V)	別ケーブルにより供給
6.VCCI(+5V)	別ケーブルにより供給
7.GND	1.GND
8.-DB0	2.-DB0
9.GND	3.GND
10.-DB1	4.-DB1
11.GND	5.GND
12.-DB2	6.-DB2
13.GND	7.GND
14.-DB3	8.-DB3
15.GND	9.GND
16.-DB4	10.-DB4
17.GND	11.GND
18.-DB5	12.-DB5
	13.GND
19.-DB6	14.-DB6
20.GND	15.GND
21.-DB7	16.-DB7
22.GND	17.GND
23.-DBP	18.-DBP
24.GND	19.GND
	20.GND
	21.GND
	22.GND
	23.RESERVE
	24.RESERVE
	25.OPEN
25.TERMPWR	26.TERMPWR
	27.RESERVE
	28.RESERVE
	29.GND
	30.GND
	31.GND
26.-ATN	32.-ATN
	33.GND
	34.GND
	35.GND
27.-BSY	36.-BSY
28.GND	37.GND
29.-ACK	38.-ACK
	39.GND
30.-RST	40.-RST
	41.GND
31.-MSG	42.-MSG
32.GND	43.GND
33.-SEL	44.-SEL
34.-I/O	50.I/O 注意すること
	45.GND
35.-C/D	46.-C/D
36.GND	47.GND
37.-REQ	48.-REQ
38.GND	49.GND
39.VCCI(+5V)	別ケーブルにより供給
40.VCCI(+5V)	別ケーブルにより供給

X68030以外の機種の場合はケーブルを購入するだけでいいので，内蔵は比較的簡単にできる

写真上のようにアクティブターミネータを装着することで安定性が増す

倍密MOは買い時か?

Hamazaki Masaya 浜崎 正哉

● MOドライブを使用する環境 ●

X68030でMOを使用する場合はなんの問題もない。SCSI-2準拠のMOを買ってくれば，だいたいのものはつながるし，そのまま使用できる。X68000シリーズの場合もSCSIインタフェイスがあれば，とりあえず接続は可能だ。しかし，デバイスの識別コード(inquiry data)で問題が生ずるので，事実上，X68000で(シャープ，ソニー製ドライブのMO)を使用するときには，inquiry dataを強制的に書き換えるINQPATCH.Xが必要となることを覚えておこう。INQPATCH.Xが手に入らないときには，ハードディスク互換モードにすれば使用することができるが，メディア交換ができないため利用価値が半減してしまう。

ということで，機種を選ぶ基準としては，SCSI-2に準拠していて，ハードディスク互換モード，MOタイプの切り替えがついていること，この2点に注目すればいい。あとは，ドライブメーカーが，ソニー，富士通，松下，オリンパスであることを確認できれば，まず問題はない。まず問題はないといっても世の中にはひねくれものもいるので，詳しそうな店員に聞いて十分な確認をとっておくか，実際に使用が確認されているものを購入するようにしよう。

そして，メーカーサポートという安心感を得たければ，ロジテックのEclaceシリーズがお勧め。きちんとX68000シリーズに対応しているし，製品ラインナップも豊富。一度カタログを覗いてみるといいだろう。

● EMO-L230 ●

さて，今回倍密ドライブとして実際に触ることができたエレコム「EMO-L230」。性能的には，バッファ容量237Kバイト，ディスク回転数3,600rpmと高速タイプの単密MOの倍密版といえるもの。X68030上からいろいろ実験してみた結果，読み込み速度はさほど変わらなかったが，やはりディスクの構造上の違いから，書き込み速度にかなりの差が出た。ちょこちょこSX-WINDOWでいじってみてもなんの問題もないし，バックアップもストレスなくサクサク取れる。

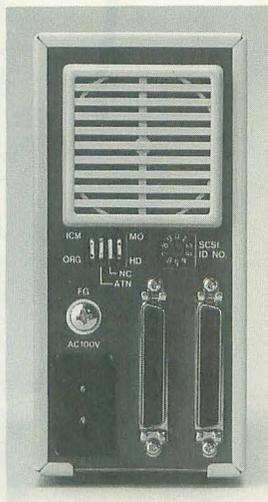
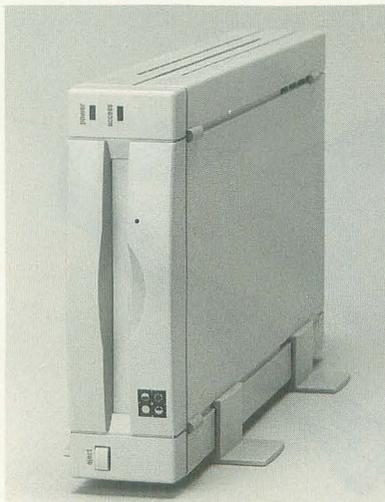
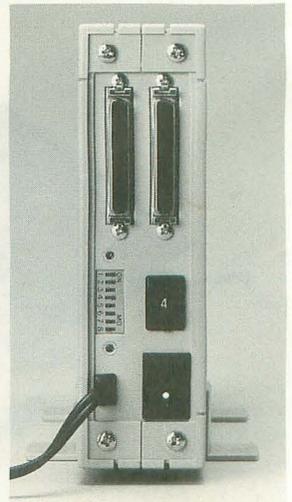
ただしMOにメディアを差さずに起動した場合，CDDEV.SYSにCD-ROMとして認識されてしまう点が気になった。ドライブが悪いのか，ドライバに原因があるのかちょっと調べきれなかったが，もしもこのドライブとCDDEV.SYSを使うことがあれば注意してもらいたい。

お約束。ディスクのシャッターを開けてみた

● 結局，倍密MOは「買い」か? ●

あくまで大容量デバイスのバックアップ用，ちょっとしたデータの持ち運び用のデバイスとしては，間違いなく「買い」といえる。まあ，すでに市場に出回っている大半のMOドライブが倍密ということで，買わざるをえない話もあるが，損はしないはずだ。実売価格も8～10万円くらいだし(ちなみにオリンパスドライブを使用した製品はちょっと高めらしい)，リムーバブルメディアの購入を考えている人は，第1候補としていろいろ調べてみるといいだろう。

●128Mバイト● ICM MO-4120

倍密ドライブに押されて，だんだん影の薄くなっていく128Mバイトドライブ。メディアの大容量化，高速化が進むいまとなっては，ディスク回転数3,600rpmの高速タイプでないと，結構つらい。それに普及率しだいではZipドライブのほうを思い切って購入するのもいいかも

●230Mバイト● エレコム EMO-L230

〈基本性能〉
- キャッシュメモリ：237Kバイト
- ディスク回転数：3,600rpm
- 平均シークタイム：30ms
- データ転送速度：3.4Mバイト/s(非同期)
- ドライブメーカー：富士通

価格 168,000円(実売80,000円前後)

4倍速CD-ROMでも使ってみるか

Sugimura Akira 杉村 晃

● X68000のCD-ROM事情 ●

なんだか世の中では4倍速ドライブが標準となり，倍速ドライブが駆逐されているようですが，我らX68000でのCD-ROM事情は，半年前からあまり進歩していません。そのような状況で，はたしてX68000でCD-ROMを使うために4倍速ドライブを購入すべきなのでしょうか。まずは，X68000でCD-ROMを使うために必要なことを復習しながら，考えてみましょう。

現在，X68000で使えるCD-ROMドライブは，SCSI-2に準拠していて，計測技研の「CD-ROM DRIVER ver.2.1」に収録されているドライバCDDEV.SYSでサポートしているものといえます。このドライバを使えば，ISO9660フォーマットのCD-ROMが読めるようになります。

ちなみに，CDDEV.SYSでは，

・東芝製のドライブ
XM3301/3401/4101/5901
・ソニー製のドライブ
CDU-561/55S
・PLEXTOR製のドライブ
・松下寿製のドライブ
・ナカミチ MBR-7

以上のドライブで動作が確認されているようです。あと，データを読むだけならNECドライブでも可能，ということはOh!X1995年1月号でも紹介しましたね。

そして，CD-ROMのなかには2.4倍速とか，4.4倍速などを宣伝文句にしているものもあります。うちのドライブは速いんだよ〜といいたいのでしょうが，これはメーカー固有の機能であって，CDDEV.SYSではサポートされていません。つまり，正式サポートされていないX68000では無用の長物なのです。普通に使うぶんには小数点以下を切り捨てた2倍速，4倍速のドライブとして使用することになります。やっぱり速いドライブがいいなあ，ということでちょっと奮発して高めのCD-ROMドライブを購入しても，無駄な投資となりかねないことを覚えておきましょう。

確かに起動時にコマンドをさくっと送ってやれば，最高速でガンガンにアクセスできるようです。実際にいくつかのドライブのコマンドは解析されているようなので，そういった特殊な情報を入手できるのなら購入を検討するのもいいでしょう。

● CD-ROMの魅力を探す ●

CD-ROMは，データ解析に命をかけるような，能動的にマシンを活用するのが好きな人にとっては，そこそこ興味深い周辺機器でしょう。質はともかく，さまざまなデータが氾濫していますからね。あとはCD-ROMでなければできないこと(PhotoCDの作成など)を見つけて，導入に踏み切るしかないでしょう。どちらにしてもしっかりとした動機と，できることをわきまえて購入しないと後悔することになってしまいます。

本来ならば，きちんとメーカーがサポートし，情報の公開，ソフトが充実してくれば状況ががらりと変わるのでしょうが……いまさらだよなあ。

● 4倍速ドライブは必要なのか ●

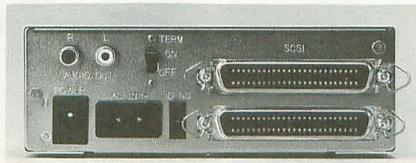
ということで，実際に4倍速ドライブのLK-RC504(松下ドライブ)を使用してみました。オーディオデータも問題なく再生でき，アクセスはそこそこ快適，問題らしい問題はありません。唯一，ディスクのローディング時間が，LCD-200より若干長めなのが気になる程度です。

そして，4倍速という名前のとおり，倍速ドライブよりだいたい2倍ほど速いです(すごく間抜けな表現ですがほかにいいようがない)。操作している間，アクセスによるストレスもそれほど感じません。これくらい軽いと，ちょっとくらい多めに投資しても4倍速ドライブを購入しようかな，という気にもさせてくれます。

結論としては，とりあえずデータが読めればいいや，とアクセス速度にこだわらない人は，中古ショップなどで安い倍速ドライブを探すようにし，倍速ドライブ＋数千円の投資が気にならないのなら，4倍速ドライブを選ぶようにしましょう。

●4倍速● 松下 LK-RC504

〈基本性能〉
・キャッシュメモリ：256Kバイト
・アクセスタイム：200ms
・データ転送速度：600Kバイト/s
・ドライブメーカー：松下寿
価格 39,800円(実売20,000円台後半)

●倍速● ロジテック SCD-200

編集部内で他機種用CD-ROMのデータ(主にゲーム機)を覗いて遊ぶために使用されている倍速ドライブ。定価が19,800円(本体のみ)と結構安い。ドライブは速ければ速いほどいいが，X68000で使う限り倍速ドライブでも十分実用になるだろう

画像入出力環境の強化

Sakamaki Katsumi 坂巻 克巳

● お絵描き職人の強化道具 ●

512×512ドット，65536色のグラフィックをもつX68000。現在，フルカラーが当たり前といった状況ですので，ちょっとだけ見劣りするスペックとなってしまいました。それでも最近では，ソフトウェアの整備により，かなりいろいろなものが表現できるようになっています。先月号のCGAコンテストの入賞作品を見ても，標準システムだけかなりの表現が可能であることがわかるでしょう。

そこで，やはりあると便利なのが画像入出力装置です。ここでは，画像入出力，特に一般的なプリンタ，スキャナ，タブレットといったところを紹介します。映像となるとまた別の周辺機器が必要ともなりますが，そのへんはDōGA CGAアニメーション講座ver.2.50でレビューを行っているので，そちらを参考にしてください。

・プリンタ

まずは出力装置。最近のパーソナルプリンタの定番といえば，やはりカラーインクジェットプリンタです。そんななかでも，現在購入するとなれば，エプソンのマッハジェットシプリンタか，キヤノンのカラーバブルジェットプリンタしかないでしょう。ランニングコスト，印字品質ともに満足のいくものですし，製品ラインナップの充実ぶりも嬉しいかぎりです。

MATIERver.2.1では，BJC-600J(360DPI)，BJC-820(360DPI)，MJ-700V2C(360, 720DPI)，DeskJet505J(300DPI)といったカラープリンタに対応しています。BJプリンタであればBJC-600J/820の設定で，MJプリンタであればMJ-700V2Cの設定で問題なく使えるでしょう。写真のBJC-400JもBJC-600Jモードでちゃんと動いています。

そして，A4サイズ360DPIのプリンタであれば，どれも性能的には似たようなものなので，わざわざ最新機種を買わずに，型落ち機種を狙ってみるのもいいでしょう。ちなみにX68000で使う場合は，ソフトがサポートしていないので，プリンタ内蔵のフォントはほとんど役に立たないことを覚えておきましょう(印刷ソフトを自作すれば別ですが)。

・スキャナ

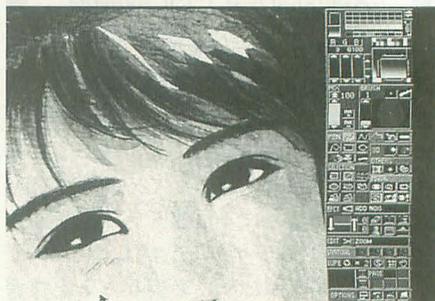
最近では，スキャナもSCSI接続するものが多くなってきました。とりあえず，SCSIであればX68000でも接続が可能であると思われますが，結局，アクセスするためにはそのスキャナがもつ専用コマンドを使用しなくてはならなりません。ということで，X68000でソフト的なサポートがちゃんと行われているものとして，シャープのJXシリーズ，エプソンのGTシリーズ(1000/3000V/4000/6000/6500/8000をMATIERでサポート)があります。

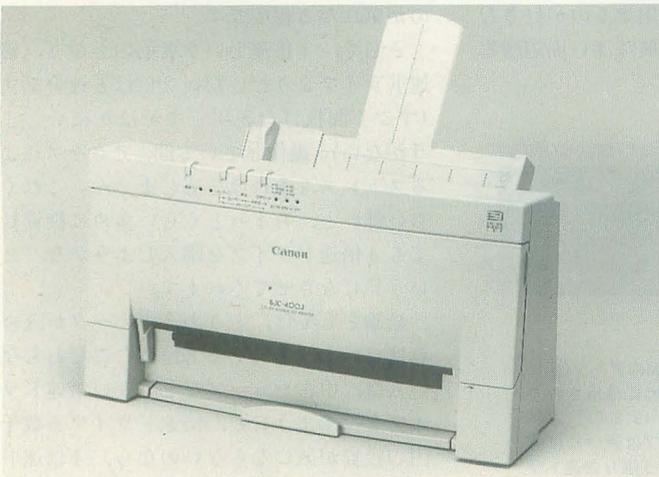
特にJX-330Xは，シャープ製の周辺機器にしては，珍しく購入意欲をそそられる製品です。A4サイズの原稿を取り込み可能で，解像度も600DPIと個人ユースであれば必要十分でしょう。SCSI接続のため読み込みが高速ですしね。ちょっと図体がでかいのが問題といえば問題ですが。

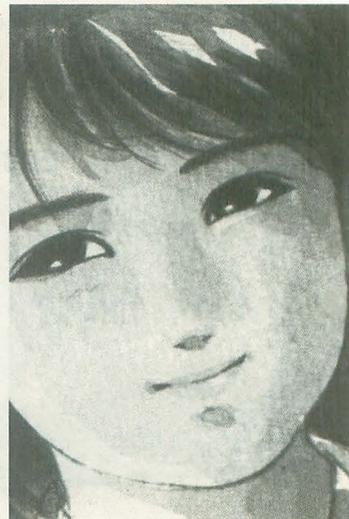
ちなみにJX-330MというMacintosh用のものもありますが，ハード的な中身はJX-330Xとまったく同じものです(実は写真のマシンはJX-330Mです)。ただ単にマニュアルとスキャニングアプリケーションが，対応マシンのものが同梱されているだけです。ちょっとだけ，JX-330Mのほうが実売価格が安いようなので，できるだけ安く手に入れたければ探してみましょう。

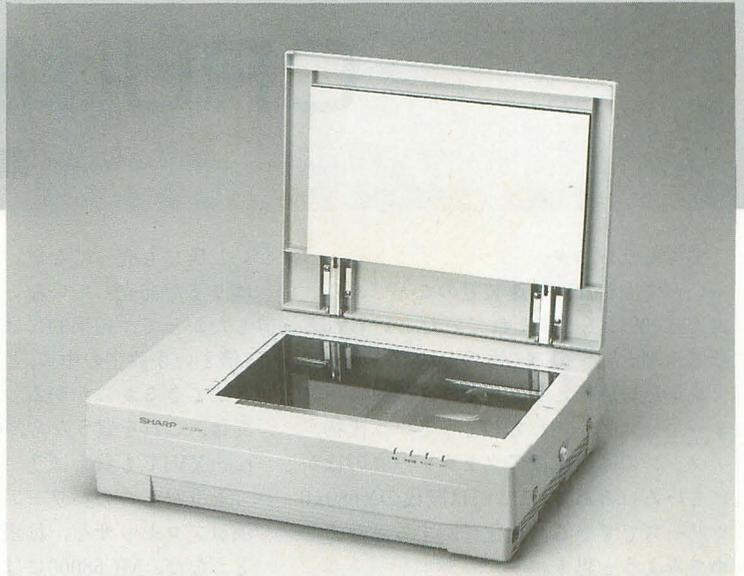
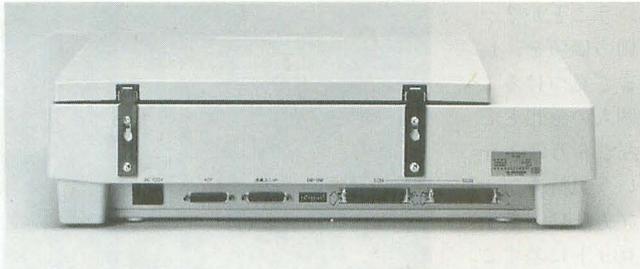
・タブレット

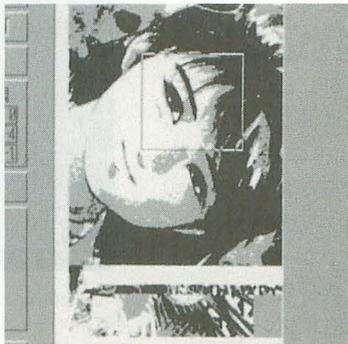
タブレットは，本誌でも何度か紹介した

MATIERver.2.1

X68000のグラフィックペイントツールの標準的ソフト。各種周辺機器への対応が嬉しい

●プリンタ● キヤノン BJC-400J

価格 69,800円(実売50,000円前後)

元画像

360DPIでの出力(モノクロ)

●スキャナ●　**シャープ　JX-330X**

価格　178,000円（108,000円）

り，ドライバが制作されたりして，結構お馴染みのNSカルコンプのDrawingSlateがお勧め。筆圧，筆の傾きまで感知する本格的なものです。残念ながらMATIERでは筆圧のみの対応ですが，それでも筆ならではの味を出すことができます。マウスよりも筆らしい表現のできるタブレットですが，やはり操作に慣れるまで多少時間がかかります

右がJX-330Xで取り込んだ画像。色ムラも少なく，結構きれいに取り込める。取り込み速度は「さすがSCSI」といいたくなるくらい速い

立体物の取り込みも結構きれい

（といってもマウスでお絵描きするのに慣れるよりは短いかな）。最初のうちはかなりイライラしますが，いったん慣れてしまえばこっちのもの。マウスでは難しかった表現も可能となります。

・レーザープリンタ

最近では，グラフィック，主にモノクロドローデータの出力に大活躍のレーザープリンタが使用できます（ただし，SX-WINDOW ver.3.0上での話ですけどね）。対応機種は，ポストスクリプト（Adobe），LIPS3（キヤノン），ESP/PAGE（エプソン）などのプリンタで，高品位な印字も可能……といいたいところなのですが，できることは，コード印字とビットイメージのみ。つまり，X68000上では，レーザープリンタといえども，普通のプリンタとできることはさほど変わらないのです（多少はきれいだけど）。それに，アプリケーションもない現状では，とりあえず購入を勧めることはできないでしょう（どうしても買いたいのなら止めませんが……）。いまのところは様子見ってところですか。メーカーからのきちんとしたサポートを待つしかありません。

●タブレット●　**NSカルコンプ　DrawingSlate**

価格　74,800円（実売50,000円台）

●レーザープリンタ●　**キヤノン　LASER SHOT**

U氏御用達のLASER SHOT。ちなみに中古で購入

CPUは加速する

Taki Yasushi 瀧 康史

アクセラレータ。

いわずもがな，加速装置のことである。古くは009にまで積まれていたといわれるほど，割と歴史のあるものなのだ。既存のマシンを買い換えることなく，簡単にパフォーマンスアップしてしまうという代物。

マシンの高速化はアクセラレータのほかにもいろいろある。ここでは歴代のX680x0シリーズでできる高速化に関する話をまとめてみようと思う。

● X68000初代 ●

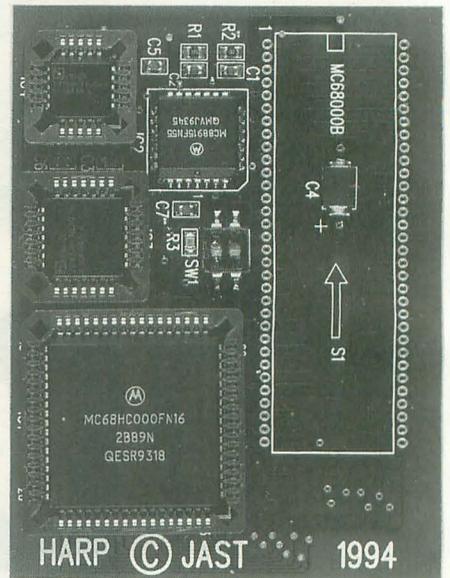
もはや御老体ともいえるこのマシンだが，わずかに高速化の余地が残されている。

利用できる手段は2つ。まずは，HARP。これは通常10MHzのバスライン上に，20MHzのMC68000を搭載したもの。搭載MPUがMC68000なのでソフトウェアの互換性は完璧。20MHzじゃ，2倍じゃん！っていいたいところだが，バス調停の都合上，実質平均1.1倍程度にしか速くならない。一応，ジャスト側はER10Sという専用メモリボードを使用することにより，1.4倍程度まで速くすると主張。原理を聞けば確かに1.4倍ぐらいになりそうな仕様なのだが，なにせ評価するボードは手もとにはないし店にも売ってない。ということはやっぱりまだ発売していないということか？したがって，結果的には定価の価値を，1.1倍という数字の中に，どこまで見つけることができるかにかかる。私個人の感想としては，これを買うなら，中古で6万のCompactXVIでも買えば？　という気持ち。

もうひとつのパワーアップ方法は，数値演算プロセッサを，拡張スロットに差すこと。ただ，MC68000にはコプロセッサ接続はできないため，あくまでバスライン接続になる。バスライン接続のMC68881/2はfloat3.xで一応は生かせることになっているが実際はたいした結果は得られない。ちなみにHuman68k ver3.0に添付されたfloat3.xより，68882に対応している。その前のfloat3.xでは68881のみの対応だ。

数値演算プロセッサによるパフォーマンスアップを望むなら，「実数を使ったプログラム」を，gccにて-m68881オプションを指定し，再コンパイルしたときに有効になる。ただ，「実数を使ったプログラム」が，世の中にはあまり存在していないのが実情。

初代マシンのスロット2つは，かなり貴重である。なぜならメモリもHDDのためのSCSIも，どちらも拡張スロットにつけなくてはならないからだ。ツクモのSCSIメモリボードを利用すれば，もう1スロットに68881ボードを搭載できるが，そこまでしてまで得る恩恵はないといっていい。

初代ユーザは買い換えの時期か？

HARP

10MHz機用のアクセラレータとして登場したHARP。専用メモリボードの使用でその性能を発揮するといわれているが……
ジャスト
価格29,800円

● ACE,EXPERT,EXPERTⅡ,SUPER ●

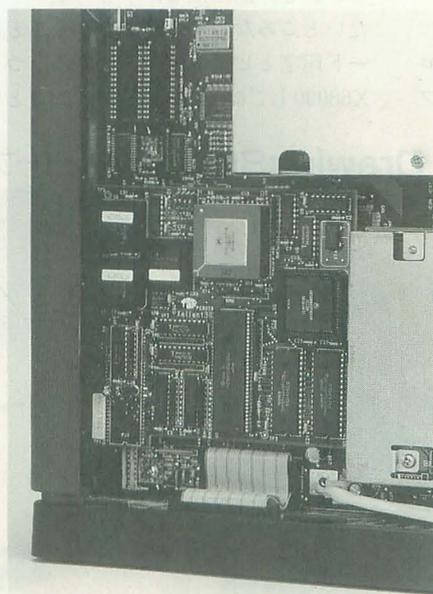
今回別のページで紹介したXellent30sが搭載できるという点はかなり大きい。製品の詳細は74ページのレビューを読んでいただきたい。

実装によりに10MHz時の使用において，だいたいXVI未満程度の速度を引き出す。速度的には「まあまあ」で少々高めの価格設定かもしれないが，速度以外に「030MPUである」という喜びと，「68030＋68882のソフトが動く」という喜びがあるだろう。

実数演算は「コプロセッサ接続の」68882が搭載されているため，gcc -m68020 -m68881オプションで実数演算はフルに高速化できる。

HDD I/FはSUPERを除いてSASIマシン。したがってSUPER以外は初

Xellent30(s)

キャッシュON時で1.5倍，ローカルRAMを使えばかなりの性能を引き出すXellent30(s)。ちなみに写真のXellent30sは68030に差し替えられている

東京システムリサーチ
価格　Xellent30　59,800円(実売4～50,000円)
　　　Xellent30s　54,800円(実売4～50,000円)

期型と同じで，なんらかのSCSIボードが必要。

● PRO，PROⅡ ●

一応，HARPだけは対応しているが，HARPをつけるよりも，15MHz程度にクロックアップしたほうが，高速かもしれない（個体差は大きい）。

同時代のEXPERTシリーズと比べ，スロットが多いだけのマシンではなく，なぜかカスタムチップまで違う。そのためタイミングが多少違うのか，いろんなボードが「PRO非対応」と名乗りをあげている。

実数演算の高速化は初代機と同じ程度の恩恵しかない。

● XVI ●

ユーザはかなり多いXVI。XVI用のXellent30は16MHz時の利用で，24MHz相当の速度を出す。20MHzに改造されたXVIでは，X68030未満という快適さで動き，場合によっては瞬間最大速度はX68030-25MHzを上回ることもある。そのため，XVIユーザの中ではかなりの人気商品のようである。

数値演算プロセッサはX68030モード時に「コプロセッサ」として利用可能だ。68000モード時も「数値演算プロセッサ」として利用するためのコプロソケットがある。メモリの実装は専用スロットがあるため，8Mバイトまで実装可能。純正品はめちゃくちゃ高いので，X-SIMM VIが魅力的。ただし，メモリフル実装にしたいとなると，結局スロットを1つ潰してしまうのが痛い。

040turbo

MPUパワーを追い求める人の最終目的地(?)がこの040turboだ。写真は温度計，通風口と仰々しいファンがついているU氏のワークマシン
計測技研
価格　99,000円

● Compact XVI，030 Compact ●

やっぱり実装スペースの問題上，クロックアップ程度しか，スピードアップは望めない。

Compact XVIのほうは，X-SIMM VIcで8Mバイトまで内蔵可能となっているが，これを利用すると，数値演算プロセッサの接続は不可能になる。純正メモリボードには数値演算プロセッサがついているが高い。最近では，ツクモが68882を搭載したCompact XVI用6Mバイトメモリボードを出しているので，これを利用するのもいいだろう（でもそんなに安いわけではない）。

030 Compactのほうもクロックアップはできるが，もともと小さな筐体に収めている手前，純正状態でもかなり熱を出す。そのため個人的にはあまりクロックアップは勧められない。

実数演算の高速化にはコプロセッサ68882を利用できるが，素人では分解したら，組み立てるのが多少難しいかもしれない。難が目立つようだけどX68030だからね。メモリを登載することに関しても問題ない。やっぱり強力ではあるな。

● X68030 ●

そしてX68030である。事実上，X68シリーズの最上位機種といえるマシンだ。マシンそのものが結構値がはるという欠点があるが，メモリもコプロもきちんと乗る。そしてなによりの利点は，このマシンに040turboが搭載できることだ。040turboの値段は高いものの，かなりの高速化が望める。搭載してときの速度の向上は，だいたい68030の2倍から3倍程度とみておけばいいだろう。

X68フルセットというとX68030＋040turbo＋12Mバイトメモリとなるが，ここまでくると安く買っても35万円ぐらい。なかなか手が出ない値段ではある。「ハイエンド」として存在する（いまのところは），速度的には不満のないXシリーズの最高峰といえる。

結局，クロックアップする技術はないけど処理速度がほしい，という人はアクセラレータの購入を考えればいい。多少のリスクはともなうが，きちんと使えばその実力を発揮してくれるだろう。

表　アクセラレータ性能グラフ（参考）

注）表中の数値は10MHzのX68000を100とした値

試用レポート

Xellent30s

Taki Yasushi 瀧 康史

お買い得なのかなぁ？

東京システムリサーチからX68000シリーズ10MHz機用（初代とPROは除く）CPUアクセラレータとしてXellent30sが発売されました。すでにX68000XVI用の製品が出荷されていましたが，正直な話，XVI用68030アクセラレータXellent30は，レポートを行った紀尾井氏も私も，それほどお得なものとは思っていませんでした（ちなみに２人ともクロックアップX68030ユーザー）。ところが買った人はみんな満足しているようですし，なかには「これでX68030はいらないよ宣言」をしている人もいます。そういうもんなのかなあ？ と思わされる出来事でした。

本当にX68000のプログラムを高速に実行したいならX68030を購入するのが，もっとも確実ですし，それだけの成果も上がります。ただ，あまりお金をかけたくないとか，いま使っているマシンに愛着があるという人もいるでしょう。値段的にはCompactやREDZONEの買い足しを考えるのとそれほど変わりませんし，コストパフォーマンスも大差ないと思われます。しかし選択肢が増えるのは確かによいことでしょう。

というわけで，このACE，EXPERT，EXPERT II，SUPER用の68030アクセラレータXellent30sも，あくまで「私は」あんまりお得な雰囲気はしてないんですけれど，すでに購入しているユーザーは案外満足しているかもしれません。

それこそ人それぞれって話でしょうけど，レポートする立場としては，もうちょっと一般的ユーザーの気持ちをわかる必要があるかなぁと思ったのでありました。

基本スペック

基本的なハードウェアスペックは，XVI用68030アクセラレータXellent30とほぼ同じ。ソフトによる認識や，Human68k ver.3.0が必要（SX-WINDOW ver.3.1に同梱して売られている）ドータボード上にMC68EC030-40MHzが搭載されていて，このMPUがMPUクロックの２倍速で動きます。

これに数値演算コプロセッサとしてMC68882が非同期で33MHz動作します。MPU用のローカルRAMとして256KBのSRAMを装備。あらかじめついていた68HC000はこのドータボード上に付けることができ，ソフトウェアによって２つのMPUを68EC030/68HC000のどちらかに切り換え可能です。

Xellent30s

ノーマルのXVIはMPUクロック16.6MHzなので，MC68EC030は２倍速で33.3MHz動作します。しかし，多くのXVIユーザーはいわゆる高速改造をして24MHzで駆動するケースが多いのが現状です。クロックアップマシンに対しては「保証はしないができる限りの措置はする」というのがメーカー側の姿勢なのか，20MHzまでクロックダウンすれば，68030-40MHzとして利用することができました。

今回のマシンターゲットは10MHz機なので，２倍速で20MHz動作します。そこで搭載しているMPUはMC68EC030-25MHz品になり，値段が５千円安くなっています。ちなみに，ソケットには最初から68030用のものが装備されているので，差し替えればMMUを使ったプログラムも動作するはずです（68030動作は確認したが，NET BSDの動作は未確認）。

コプロセッサはMC68882-33MHzと非同期ですので，実数演算に関してはXVI用のXellent30と同じ速度で動作します。

残念ながらどのぐらいまでクロックアップ機に対応しているか，調査する環境がありませんでした。ボード自体がX68000本体が耐えるクロックで動作しても，システム全体が動くとは限りません。このへんは個体差が大きいところですが，高速型CPUへの交換などで多少改善されることもあるようです。

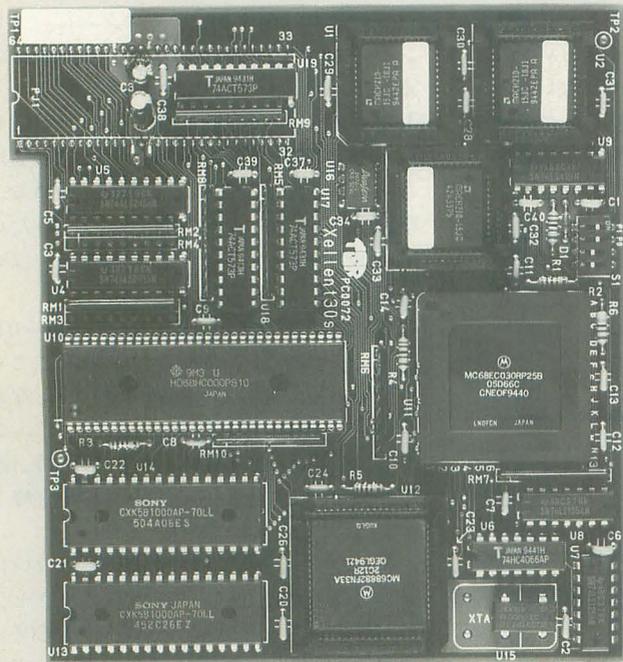
パフォーマンスアップ

さて，このアクセラレータを搭載することによって，パフォーマンスはどのくらいまでアップするのでしょうか？ だいたい体感的には５割増しくらいといったところでしょう。

高速化において，Xellentは下の３つの利点があります。

1) MPUが２倍速の68030
2) MPU内蔵キャッシュ
3) コプロセッサが使える
4) ローカルメモリが256Kバイトある

まずはCPUが68030になったということにおける高速化です。68000と68030では，同じ命令での実行時間は異なります。これらが如実に現れるのは，割り算，掛け算，シフトなどの演算です。

これらは68000では「遅すぎ」といわれていて，平均で60クロックぐらいの実行時間を有します（１クロックはMPUクロックの逆数）。もちろん，そのほかの命令も高速化されてはいます。68000では１命令の最短命

令実行時間は4クロックでしたが，原始的なパイプラインが効いたのか，68030では2クロック程度になっています（ただし32ビットバス時）。この程度というのは前後にある命令で多少異なるからです。

ところが，以前HARPのレポートのときに説明したとおり，MPUの実行時間(クロック)の大半は，MPUの外にアクセスする時間に大きく左右されてしまいます。したがって結果的には，CPUが68030になっても「キャッシュをオン」にしないかぎり，1個の命令の実行時間は極端には変わらず，ワーストケースでは68000のバスラインにあわせたことにおける調停時間が影響し，逆に遅くなってしまいます。

そこで2のMPU搭載のキャッシュが有効になります。68EC030のオンチップキャッシュ256バイト×2本（命令/データ用）のキャッシュは原始的なものではありますが，あるとないとでは大違いです。キャッシュにさえ載れば，1)の68030の命令速度の利点もそのまま生きてきます。

MC68030の利点のひとつに，コプロセッサの接続があります。MC68000ではコプロセッサを使用することはできず，一般のバスラインを通してコプロセッサを接続していました。Xellentを利用すると，これがコプロセッサ接続になるので，実数演算はかなり高速化されます。具体的にはfloat4.xを利用すればいいのですが，いろんなところで説明したとおり，FLOATファンクションは遅いため，実際ではGCCなどでのコンパイルし直しでしか恩恵はほとんど得られません。

コプロセッサ自体，計算に実数，つまり小数点が現れるようなプログラムでないと，恩恵は得られないわけなので，購入したすべての人が恩恵が得られるか？　という話になると，これはまた別問題になります。特に演算能力の要求されるXL/Imageの030オブジェクトやREND030をそもまま使用できる，というのは大きなメリットですけれども。

したがって，高速になるかどうかは，ほぼ，2本のキャッシュにかかっていることになります。4)のローカルメモリは，利用すると速くなるイコール，「使わないと速くならない」ということなので，既存のソフトの恩恵は，結局2本のキャッシュの双肩にかかっているわけです。

ではいったい，既存ソフトはどのぐらい速くなるのでしょうか？　まずはSX-WINDOWを普通に使ってみました。いつもX68030を利用しているからってのもあるとは思うのですが，感覚的には「速くなってるのかなぁ？　なってるんだろうなぁ？」程度の感じです。Xellentを使い始めたそのときにはなかなかわからないけれど，使うのをやめたとき(68000モードにしたとき)に重たさを感じるので，やっぱり速くなっているのでしょう。

シャーペンも使い方次第で重くなるので，一概にはいえませんが，激しいこと(フォントマンでベクトルフォントを使うなど)をしなければ十分実用レベルです。現にこの原稿はいまX68000SUPER(10MHz)＋Xellent30sで書かれていますが，長い段落をインサートモードでインライン変換しなければ，重さは感じません。

1月号のXellent30のときに行われた，SX-WINDOWの画面表示速度ベンチマークでは，

X68000SUPER	10MHz	15119
Xellent30s	20MHz	10170
X68000XVI	16MHz	9774
Xellent30	33MHz	5885
X68030	25MHz	3871

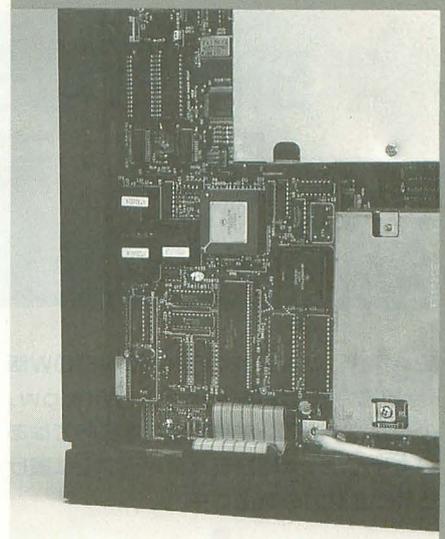
という具合ですから，確かに速くなってはいるようです。「だいたい」のベンチマークですが，表示速度はXVI未満といったところでしょう。

ゲームはどうでしょう？

SUPER STREET FIGHTER IIを試しにプレイしてみました。10MHzではゲームにならないとまではいきませんでしたが，ところどころで重さを感じたものです。こちらのほうも目に見えて速くなりました。もちろん，68030モードでPCM多重再生できます（必ずCACHE ONで実行すること）。それでも，ディージェイステージは多少重くはなりますが，それでも前に比べたら，十分問題ない範囲の速度です。

ローカルRAM

速度的にはXVI未満というのがだいたいの感じだと思います。030なのに？　という感覚が多少ありますが，以上の理由でこれはしかたありません。

そこで，もっと有効に使おうというのがローカルSRAMです。このローカルSRAMはMPUに直結されているため，バスマージンが少なくなり，高速アクセスが可能となります。添付ソフトにloadhigh.rというソフトがあります。このソフトは，プログラムをこのローカルメモリに転送して実行するプログラムです。DMAが届かないメモリなので，PCM8などは常駐できても音が鳴りませんが，グラフィックローダなどを

転送して展開を高速動作したり，LHAなどを転送して高速動作したりすることはできます。このときの実行速度はXVI以上です。

しかしながらローカルメモリは256Kバイト。1MビットSRAMという高いRAMを使っていることは確かなのですが，足りなくて困るときがときどきあります。512Kバイトあれば FSXを入れられますし，せめて増設できれば……うれしかったのですがねぇ。

まとめ

XVI用のXellent30の場合，20/40MHz利用時は瞬間最大速度はノーマルのX68030を上回ることがありました。そういった背景上，「もうX68030はいらない」といえる速度が楽しめ，ユーザーの大半は満足しているという結果に至ったのでしょう。

今回のXellent30sはどうでしょうか？　鍵は256KバイトのローカルSRAMにあるため，買った人間がどの程度満足してくれるかはまだ未知数です。確かに，ローカルRAMをうまく使うようにシステムを組めば，多くの局面でかなり体感速度を上げることができます。ローカルRAMを効率的に利用できるマネージャがもう少し強力になれば多少変わるでしょうか？　それ以外でも最近のxgcc＋libcをうまく利用し，ツールをコンパイルしなおせば，まだまだ行けそうな気がしますが，普通に使っての速度は，XVIの16MHz未満というのが正直な感想です。

10MHzマシンユーザーは，X68030を買おうか揺れている状態だと思います。Xellent30sはどの程度，ユーザーの目に止まるかは少し楽しみであったりします。

Xellent30s　59,800円（税別）
東京システムリサーチ　☎0425(28)1824

学研統合電子辞書 for SX-Window
国語・漢和辞書/英和・和英辞書

Sugimura Akira 杉村 晃

簡単操作，楽々動作のSX-WINDOW版「学研統合電子辞書 for SX-Window」の登場です。SX-広辞苑ほど本格的ではありませんが，気軽にデスクトップ上に置ける軽さが魅力の製品です。

電子辞書です

今回，TAKERUより「学研統合電子辞書 for SX-Window」が発売になりました。同タイトルのWINDOWS版からの移植であり，内容的にはまったく同じものですが，SX-WINDOW版では国語・漢和辞書と英和・和英辞書に分けられてリリースされています。

両辞書とも9,800円(税込み)と軽い気持ちで手に入れてみようかな，と思える価格です。いっぺんに購入するのはちょっときついかもしれませが，まず，どちらか片方だけ購入し，気に入ったのならもう一方を思い切って買ってしまうことをお勧めします。

インストール

基本的にこのソフトはハードディスクにインストールして使用することを前提にされています（当たり前か)。

最終的に国語・漢和/英和・和英辞書の両方をインストールした場合，ハードディスクに約7Mバイトの空きが必要になります。注意してくださいね。

そして，ディスクにはハードディスクへのインストーラ用バッチファイルが付属していますが，操作に関して多少面倒な部分がありますので説明書をよく読みましょう（私は操作をミスってメーカーに迷惑をかけてしまった)。慣れた人なら，実行ファイルを手作業で解凍して，生成されたファイルをサクサク同じディレクトリに放り込めばOKです。ひとつだけ，ファイルを連結する必要があるものもありますが，詳しくはインストール用のバッチファイルをみてください。よくわからなかったら素直にバッチファイルを使いましょう。

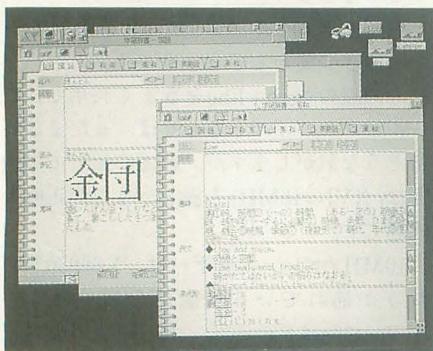
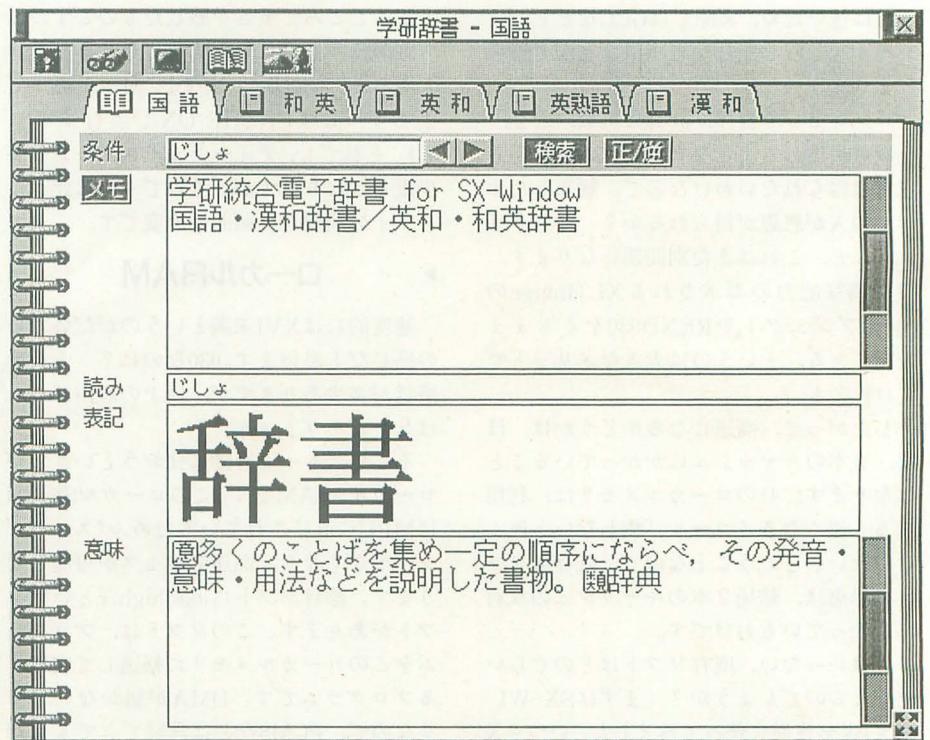
検索方法

特に知らなければならないこと，特殊な操作は必要ありません。条件と書かれたエリアに調べたい言葉を直接入力するか，クリップボードを介して単語を置くかして，検索ボタンをクリックするだけです。すると，ほとんど待たされることなく検索結果が表示されます(X68030使用時)。その単語がなかった場合は，タイトルバーに“辞書にありません”と表示されます。

また，検索結果に表示される単語をクリップボードに放り込むことで，さらに検索できますし，別の辞書で検索もできます。

検索方法も順方向検索だけでなく逆方向検索(逆引き)もサポートしているし，検索単語を16個まで登録できたり，複数の辞書を起動することができます。

以下，簡単にですが，それぞれの辞書ごとで設定できる検索条件を説明していきます(カッコ内の数値は，その辞書に登録されている予約語数です)。

●国語辞書(約39,000語)

TAKERU ☎052(824)2493
X68000用3.5"/5"2HD版 各9,800円(税込)

条件入力：直接入力，クリップボードからのコピー
条件設定個数：1
前/次候補：あり
逆引き：あり

●漢和辞書

条件入力：直接入力，クリップボードからのコピー
条件設定個数：最大4（音訓，部品，部首，総画数を4つまで組み合わせて検索条件に設定できる。直接文字を指定する漢字，区点，SJIS，JISコード検索は1つ）
前/次候補：あり(SJISコード順)
逆引き：なし

●英和辞書(約42,000語)

条件入力：直接入力，クリップボードからのコピー
条件設定個数：1
前/次候補：あり
逆引き：あり

●和英辞書(約53,000語)

条件入力：直接入力，クリップボードからのコピー
条件設定個数：1
前/次候補：あり
逆引き：あり

●英熟語辞書

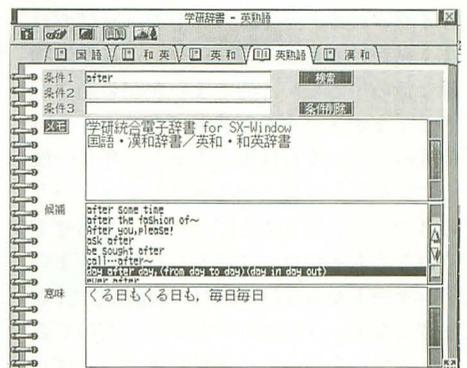
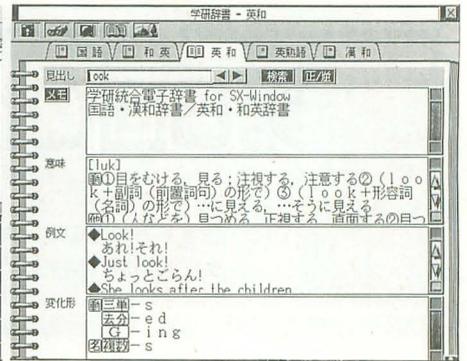
条件入力：直接入力，クリップボードからのコピー
条件設定個数：3
前/次候補：あり
逆引き：なし

操作感覚

検索方法の解説でも触れたとおり，基本的にこの学研辞書は，クリップボードを介して単語の検索を行います。学研辞書内であれば，単語をマークして検索ボタンを押すか，別の辞書を選択するとサクサク検索してくれます。特に解説書を読まなくても，何度か単語を入力して検索しているうちに自然と身につきます(漢和辞書の条件設定はちょっとややこしいけど)。さらに，環境設定によって「クリップボード文字列更新時に自動検索」といったことをさせることもでき，文書を編集しながら，簡単に辞書検索できるという便利な機能もあります。調べたい文字列をシャーペンなどからちょこっとコピーすれば，勝手に学研辞書が検索をしてくれるのです。ところが，この機能は便利な反面，多少の問題もあります。それは，検索するときに学研辞書がアクティベートされてしまう点です。結果的にユーザーは検索が終わったあと，文書編集を再開するために，シャーペンのウィンドウをアクティベートし直すしかありません。この操作が結構うっとおしく感じます。

写真は英和・和英辞書パッケージの内容。発音記号，活用形など，必要な情報はひととおり網羅している。英熟語辞書では，3つの単語を組み合わせた検索が可能だ。国語・漢和辞書より1つ辞書数が多いのでちょっとだけ得した気分にさせてくれる

確かにその単語を調べる目的で，クリップボードへコピーするのですから，検索結果が見られる必要があります。しかし，文書編集では，カット，コピーというものを多用するものであり，そのたびに辞書がアクティベートされていたのでは話になりません。

ここはやはりタスク間通信をうまく使って，裏側で静かに検索してくれるモードもほしかったですね。もしくは，シャーペン専用の外部コマンドを用意してもらってもかまわなかったのですが。SX-WINDOWの世界では，文書編集を行う≒シャーペンを使うという式が成り立っていますから。

また，環境設定，フォント選択のほかのすべての機能にショートカットキーが設定されているのは嬉しいかぎりです。

検索条件を4つまで組み合わせて検索が可能な漢和辞書。表記部分には好きなアウトラインフォントを設定できる

お勧めかな？

全般的に操作感覚もいいし，X68030で使うぶんにはかなり高速に動作しているといえます。ですが，辞書自体に登録されている単語数がまだまだ少なく，本格的な使用に耐えられるとはいえない部分があります。(特に，国語辞書の用例をもう少し充実させてほしかった)。

確かに，辞書のデータ量はハンパではないので，フロッピーベースで供給するためには，かなりデータを絞らなくてはならないのかもしれません。SX-WINDOW版ポケット辞書と割り切って使えば，まあ，満足できる出来ですけど，それでは悲しすぎます。せめて予約語と用例が倍ぐらい充実していれば，評価はかなり違っていたことでしょう(結構ムチャをいっている気がするけど)。う～む，惜しい。

ということで，システムと辞書のバージョンアップをさっそくお願いしたいなあ。作るべきところはしっかり作ってあるので，今後の対応に期待したいところです。

第6回Oh!Xアンケート分析大会

Takahashi Tetsushi 高橋 哲史

今年もお約束のアンケート分析大会です。「言わせてくれなくちゃだワ」に寄せられたアンケートの集計結果はどうなっているでしょうか？ 昨年に続いて(哲)氏が担当です。

オウムにカナリヤ。ペットショップではよく見かける光景なのに……。なにかと物騒な世の中ですが，皆さまいかがお過ごしでしょうか？ そんな状況でもちゃんと時は流れ，春がくれば夏もくるし，年に一度のアンケート分析大会もきてしまうというわけです。ということで今年も「言わせてくれなくちゃだワ」に寄せられた皆さまのアンケートから500枚を無作為抽出して，さまざまなデータを集計いたしましたのでさっそく分析を始めたいと思います。しばしおつきあいのほどを。

所有機種

まずは表1，Oh!XにおけるX68000の所有者数です。500人中471人，およそ94.2%とほぼ去年と同じ数字が出ています。ということで，これからパソコンに関する質問がいくつかありますが，それは主にX68000について回答されたものです。

そのX68000所有者を機種別に分けてみます。例によってX68000を複数台所有してらっしゃる方がいらっしゃいますので，総数は587台になってます。1人平均1.2台という勘定になりますね。グラフ1をご覧ください。昨年に続いてXVI以上のマシンの割合が増えています。確実に上位マシンへの移行が進んでいるといったところでしょうか。そんななか，なぜか初代の割合が増加しています。中古で手に入れた方が多いんでしょうかね。アンケートを見てると「いつまで経っても新機種がでそうにないので，我慢しきれずX68030を購入してしまいました。いったいシャープは……。(以下略)」という方が結構いらっしゃったので，X68030が強いのはそういった原因もあるのでしょうね。本当にシャープからなんのアナウンスもないというのは，ユーザーにしてみれば不安でしょうがないというのも事実です。私もいちユーザーとして「本当に秋にはなんかあるのかー!?」と声を大にして問いたくなってしまいます。あまり夢みたいなことは望みませんが，このままパソコン市場から撤退してしまうのではあまりにも寂しすぎるのではないでしょうか，シャープさん？

表1 X68000の所有者

	人数	割合
68ユーザー	471	94.2%
非68ユーザー	29	5.8%
合計	500	

グラフ1 X68000機種別ユーザーの割合

続いて機種別所有台数の推移です。グラフ2をご覧ください。過去のデータとの兼ね合い上，2倍して1000枚分集計したものとした値でグラフ化しています。ノーマルX68000系の落ち込みに，XVI系，030系の増加がきれいに対応してるのがわかります。あと面白いのはX1/X1turbo系がやや持ち直しているところです。「なぜ，いま頃？」と思いましたが，結構中古で購入された，というパターンの方が多いようです。なん

せ秋葉原とかではX1turboZが5,000円くらいで売られているようですから(ああ，時代の移り変わりってやつは……)。あとMSX系がいきなり4位に浮上してきていますが，これは今年より「その他」の欄から，MSXが独立してでてきただけのことで，ユーザーの方は以前からちゃんといたのです。MSX教もシャープ教に勝るとも劣らぬカルトな集団ですからね(そこまでいうか)。

システム環境，ハード&ソフト

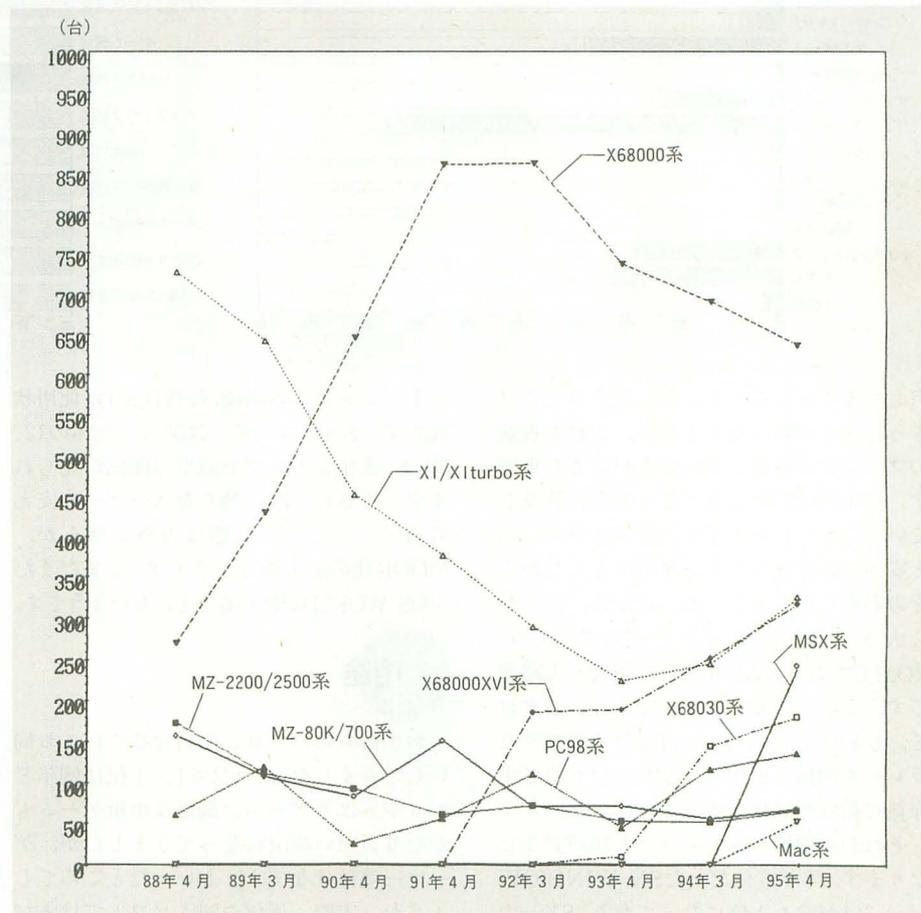
ここから先の集計結果についてはX68000所有者の471人が対象になります。複数台を所有している人はメインで使っているマシンについて答えていただきました。

では，環境について見ていきましょう。まずはメモリです(グラフ3)。2Mバイトユーザーがトップなのは去年と同じですが，その次がなんといきなりフル実装の12Mバイト！　続いて4,6Mバイトがほぼ同数で続いています。1Mバイトユーザーはほぼ壊滅状態ですね(笑)。いやあ，やっぱりメモリはあったほうがいいですよ，みなさん。「やっぱり大量メモリは正義！」と3カ月前に2Mバイトから8Mバイトにクラスチェンジした私は公言してしまうのでありました。なにをするにもメモリを気にしなくていいというのは極楽ですね。

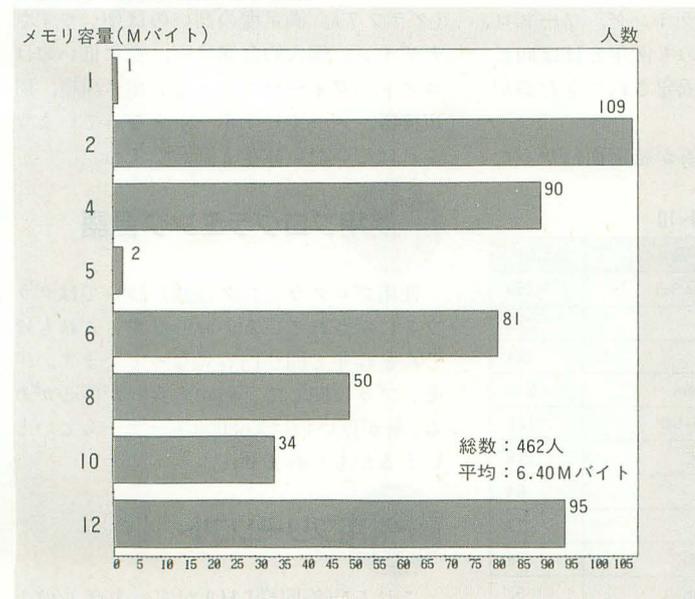
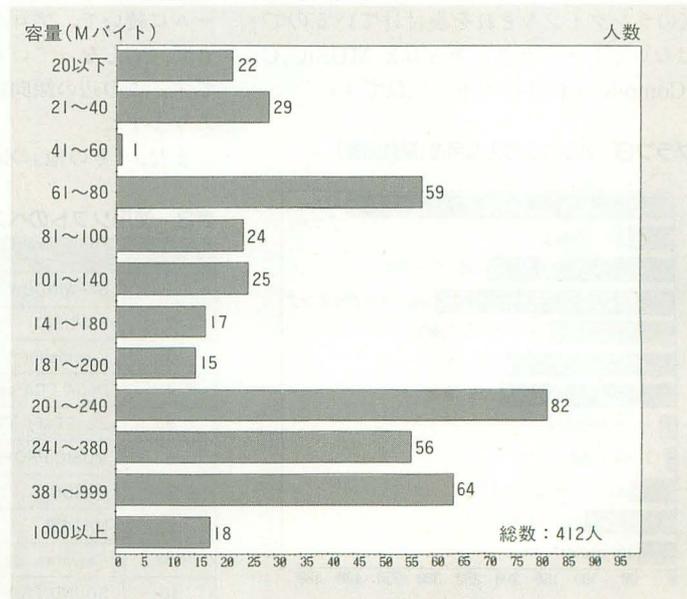
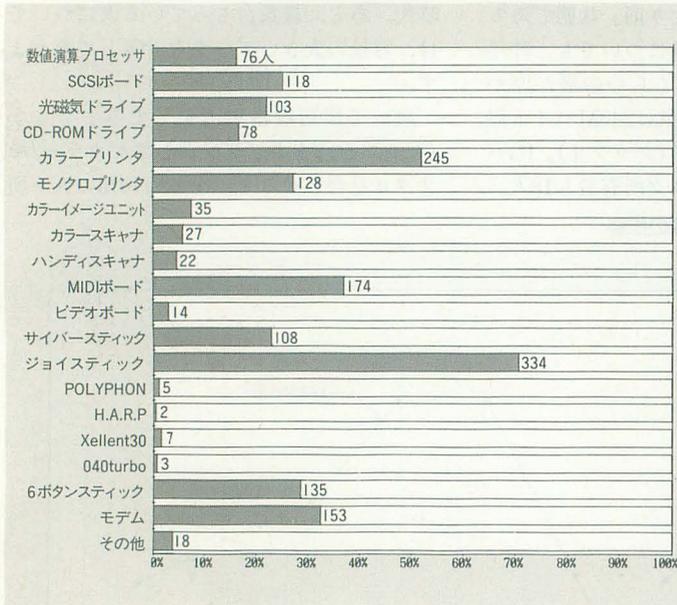
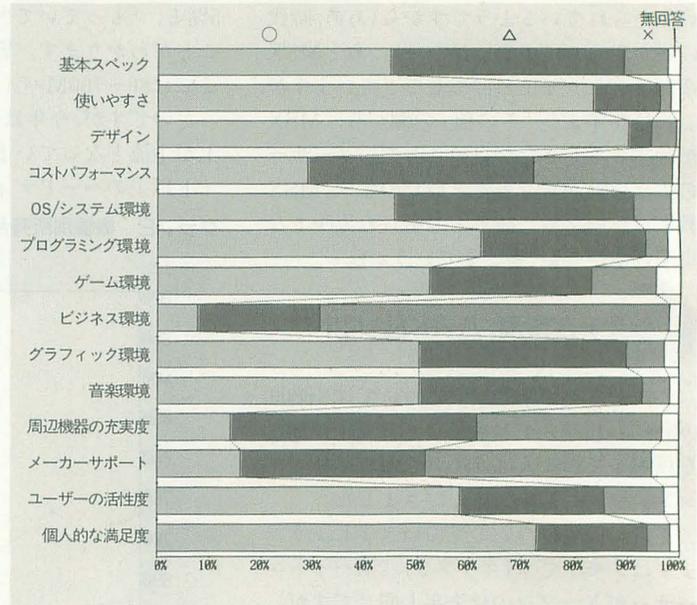
次にハードディスクの所有率です(表2)。

表2　ハードディスクの所有率

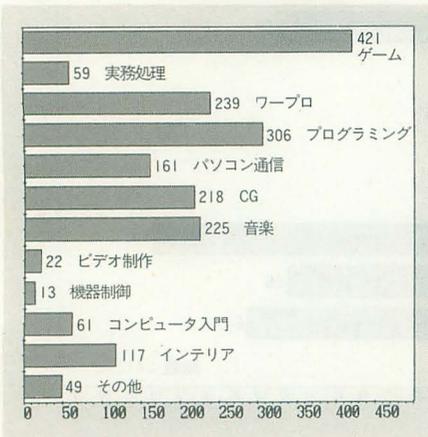
	人数	割合
所有	412	87.5%
非所有	59	12.5%

ハードディスク所有者は412人，全体の87.5%と，「もっていて当たり前」状態であることがわかります。容量についても，昨年までは80～100Mバイトくらいの層が厚かったのですが，今年は一気に240Mバイト以上が主流となっています(グラフ4)。1Gバイト以上のハードディスク所有者も18人，と昨年より増加しています。んー，大容量時代。あと，複数台もっている人については，容量の大きいデータを計算してあります。

続いて周辺機器事情を見てみましょう(グラフ5)。まずプリンタですがカラー/モノクロを合わせると，全体の70%の人が所

グラフ2　機種別所有台数の推移

グラフ3　X68000のメモリ容量

グラフ4　X68000のハードディスク容量

グラフ5　X68000ユーザーの周辺機器所有率

グラフ7　現在使用しているX68000について(○△×の3段階評価)

有してることになりました。グラフ上ではもっと多くの数になりますが，これも複数のプリンタを所有している人がいるためです。昨年の62%から比べると着実に普及していることがわかります。昨年はいろいろと安価で高性能なプリンタが出ましたからその影響なんでしょうね。ほかに，昨年から大きく変動があったところではCD-ROMですね。なんと6人から78人へ大躍進です。といっても，まだ17%くらいですけど。あと040turbo，Xellent30などのアクセラレータ関係も去年にはなかったものです。今後の動向が気になるところですね。

それから使用ソフトのベスト10が表3になります。昨年3位だったSX-WINDIOW ver.3.1が堂々1位になっており，SXerの増加を実感させますね。Easydraw SX-68Kのランクインもそれを裏付けているのではないでしょうか？　あとはZ-MUSIC，C Compilerと例年どおりの順位です。

グラフ6　パソコンの主な用途(複数回答)

そしてシステム環境，最後はS-OS使用状況です。S-OSユーザーは58人。全体の12.3%と，去年よりもやや減少の傾向が見られます。まさに一部の熱心なユーザーに支えられて……といった感は否めませんが，MOOK化の話もあるようですし，まだまだS-OS WORLDが終わることはないようです。

用途

お手持ちのパソコンの用途についてお伺いしてみました(グラフ6)。1位は例年どおりやっぱりゲーム。最近は市販ゲームもかなりお寒い状況になってきましたが，ゲーマー魂は死なず！　といったところでしょうか。実際，通信や同人ソフトではまだ熱いソフトが手に入ったりもしますし。ゲームに続いて，プログラミング，ワープロ，音楽，CGとなっているのも例年とほぼ同じです。この辺の傾向は固定されてきた感がありますね。

また，「その他」の回答が結構面白かったのでいくつか紹介させていただきます。「生活必需品」(なるほど)「改造への第一歩」(おいおい)「失恋のキズを癒す」(おーい，いっちゃってるぞー)などが面白いご意見。あと，まじめなところで「障害者のボランティア活動に使用しています」という回答もありました。いやー，X68000っていろんなところで活躍してるんですね。ちなみに「その他」の欄で一番得票数があったのは「電脳倶楽部再生機」(7票)でした。これはどう受けとっていいものやら……。

表3　使用ソフトのベスト10

順位	ソフト名	人数
1	SX-WINDOW ver.3.1	350
2	Z-MUSIC	336
3	C compiler	224
4	DōGA CGA system	210
5	Z's-STAFF PRO-68K	126
6	MUSIC PRO-68K	119
7	NAGDRV	84
8	MATIER	77
9	Easydraw SX-68K	63
10	SOUND PRO-68K	62

現状への満足度

続きまして現状への満足度です。基本スペック，使いやすさなど14項目に分けて○△×の3段階で評価していただきました(グラフ7)。満足度の高いのは使いやすさ，デザイン，個人的な満足度，逆に低いのはコストパフォーマンス，ビジネス環境，周辺機器，メーカーサポートとなっています。これはだいたい昨年と同じですね。

使用プログラミング言語

使用プログラミング言語についてはグラフ8に示されているとおりです。これもほとんど昨年と同じ内容になっています。アセンブラに関して「わからないが関心がある」層が厚いのはX68000ユーザーらしいといえるかもしれません。

常用フリーソフト

これも昨年同様LHAがぶっちぎりの1

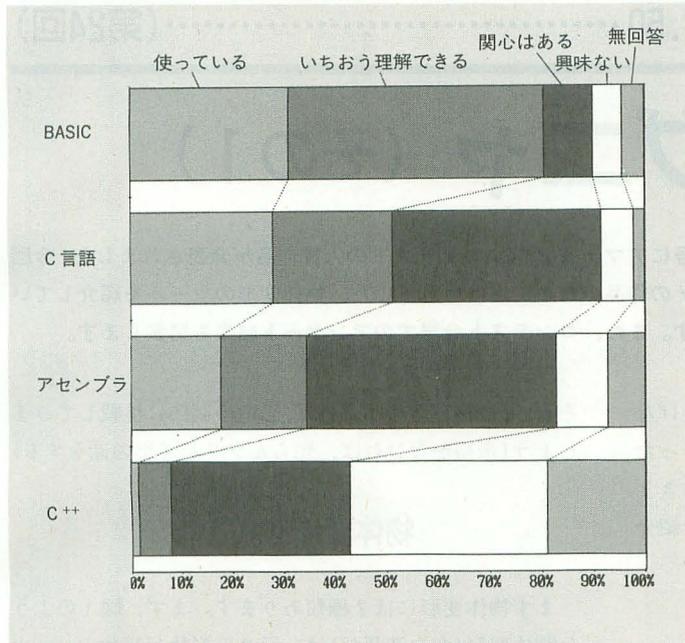
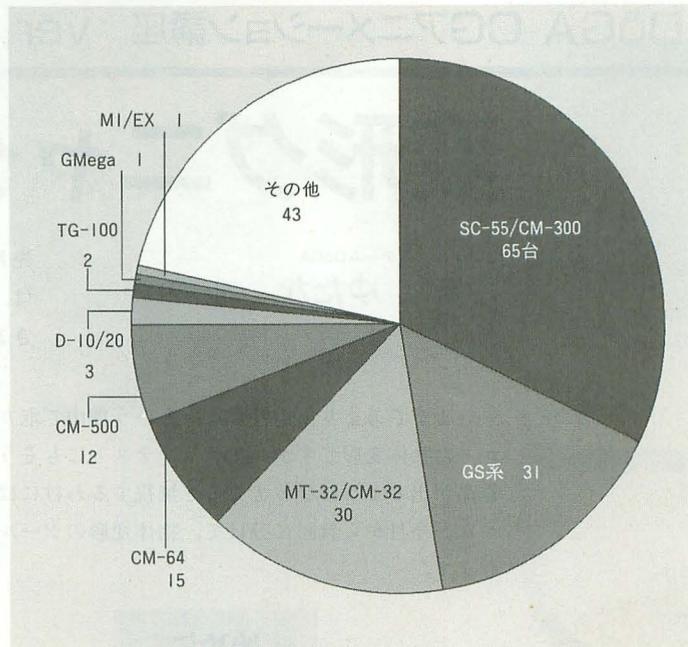
グラフ8　プログラミング言語について

グラフ9　MIDI音源機種別所有台数

位になっています(表4)。まあ通信をしていないにせよ，データの受け渡しなどするときにはLHAは必須ですからお世話になっていない人はいないでしょう(早くLH6形式にも対応してくれないかなー)。あとはだいたい例年どおりの順位なのですが，PIC2のランクインが注目されます。JPEG形式などの非可逆圧縮が幅をきかせるなか，完全可逆圧縮＆高圧縮率を誇るPIC2には頑張ってもらいたいところです(あれ，そういえばICEは？)。あと昨年10位にランクインしていたTFは半数ほどの人がMINTに乗り換えたようで，そのため票が割れランクインできなかった模様です。TFの選択肢を消してMINTに書き換えたうえで○をつけている方が何人もいらっしゃいました。なんにしてもフリーソフトの作者の皆さま方には感謝，感謝ですね。これからもぜひ頑張っていただきたいものです。あと，ランクには入ってないのですが，個人的にはJPEGEDのリアルタイム拡縮にはカンドーしました。やっぱりソフトって力仕事なのですねー。

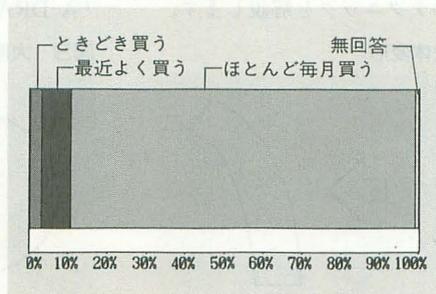
表4　使用しているフリーソフトのベスト10

順位	ソフト名	人数
1	LHA	363
2	PCM8	262
3	LZX	238
4	APIC	237
5	HAS	206
6	HLK	193
7	PI	143
8	PIC2	137
9	GCC	118
10	MXDRV	112

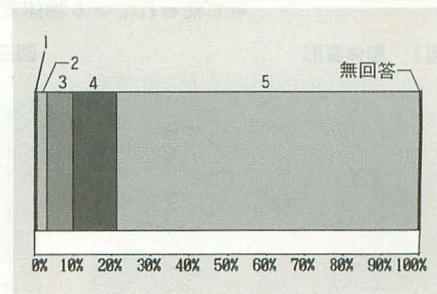
使用MIDI機器

MIDIボード所有者数も昨年より6％増え，MIDI人口も着々と増加しているようですが，皆さんが使用していらっしゃる音源の内訳がグラフ9です。もう圧倒的にSC-55/CM-300の天下になっているのがわかります。ちなみに2位の「GS系」にはSC-155,SC-88が大量に含まれていますので，結局半数くらいはローランドユーザーということができると思います。いやー，やっぱりDTMはローランドってことですかねー。ちなみにGM系のほうの内訳はTS-6GM1(7人)，WAVE BLASTER(2人)，その他(5人)となっておりました。

Oh!X意識調査

それでは最後に本誌に対するアンケートです。まずは，グラフ10のOh!Xはどのように購入されていますか？　なんと89%，444人の方が「ほとんど毎月買う」とご回答になっています。なんというか例年ながら，すごいというか恐ろしいというか驚異のパーセンテージが出てますね。新機種は発表されない，ソフトもほとんど出ない，というジリ貧の逆境のなかでのこの89%という異常な数値は，まさにシャープ教信者の真髄を表しているといえるかもしれません。さあ，この迷える小羊たちに輝ける未来はありや，なしや!?(自分もその一員なのでできればあってほしいです)。Oh!Xの必要度(グラフ11)のほうも相変わらず高いですし，いまさらながらみなさんのXシリーズにかける情熱には心打たれてしまいます。もっとも，古い読者のなかには現在のOh!X，そしてシャープに「可愛さあまって憎さ百倍」状態の方もいらっしゃいますが……(自分もその気があるのでそれはよくわかりますが)。そういう方は「毎月購入」に○をしながら，必要度には「不必要」に○をしてらっしゃるんですよね。うーん，複雑なものを感じてしまいます。

ということで第6回アンケート分析大会，これにて終了です。また来年！

グラフ10　Oh!Xの購入について

グラフ11　Oh!Xの必要度(不必要を1とした5段階評価)

変形グニャグニャ(その1)

プロジェクトチームDōGA
かまた ゆたか

先月号にアマチュアCGAコンテストの入賞作品が発表されました。今回は,その作品のなかでも使われていた,物体変形のツールを紹介していきます。また,コンテスト会場でのアンケート結果も発表します。

いままであまり実用性がないという理由で取り上げなかった物体変形ですが,CGAコンテストにもそういった作品が出るようになってくると無視するわけにはいきません。今月から数回に分けて,物体変形のツールを紹介します。

始めに

去る4月2日,東京千代田区公会堂にて「第7回CGAコンテスト入選作品上映会」が開催されました。ホールは公会堂の9階にあるのですが,開場前には入場を待つ人が,階段を9階から1階まで連なり,建物の外まで溢れてしまうほどの盛況ぶりでした。来場者は800人を超え,4時間にわたるイベントも大いに盛り上がり,好評のうちに無事終了いたしました。来場者の皆さん,ありがとうございます。

さて,今回のコンテストは下馬評どおり「A DRAGONFLY」がグランプリになったわけですが,上映会に来た方やすでにビデオを入手された方は,あの恐竜が滑らかに動き回る姿に驚かれたと思います。そして,この作品は紛れもなくX680x0,そして,CGAシステムとMATIERだけで制作されています。ということは,読者の皆さんの大部分はあのクオリティの映像を作るに足る,十分なハードとソフトをもっていることになります。でも,実際問題あんな恐竜など,いったいどうやって作るんでしょう?

そこで,今回は「A DRAGONFLY」の恐竜の作り方……といいたいところですが,さすがにあれを簡単に作れるとも,解説できるとも思えませんので,まずはあの恐竜で使われている物体変形のテクニックを解説します。それで,最後に余裕があれば,恐竜の制作に挑戦してみましょう(余裕がなければ,知らんフリして別の話をする)。

物体変形あれこれ

まず物体変形には2種類あります。まず,図1のような剛体変形(変な造語だ)は,元々の形状が複数のパーツに分かれており,個々のパーツの形状自体は変わらず,つながる部分の角度や位置が変わります。プラモデルのロボットを想像していただければわかるでしょう。

それに対して軟体変形(これも造語)は,全体が複数のパーツに分かれている必要はなく,図2のように文字どおり形が変わります。例としては,こんにゃくといった感じでしょうか。

図2の程度の変形なら,剛体変形でも細かいパーツを連ねて,それらの角度を少しずつ変化させることで表現できないこともありませんが,どうしても間接部分の継ぎ目が目立ちますし,だいいち図3のような変形はできません。

このようにひと口に物体変形といっても,いろいろな種類があり,それぞれ使用するツールが異なります。まず剛体変形は,フレームソースファイルの記述を階層化することで簡単に実現できます。これはすでに常識といえますので,ここでは解説を省略します。わからない方はマニュアルの「博士課程/構造体理論研究」(T-343)をご覧ください。

軟体変形のツールとしては,最も古いEXPOINT.X,バージョン2.Zで登場したものの,まだコンテスト応募作では使用されていないBOXTRANS.X,そして今回「A DRAGONFLY」で使用されたYAWARA.Xの3

図1 剛体変形

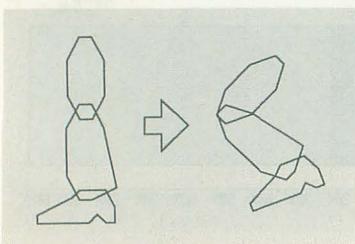



図2 軟体変形

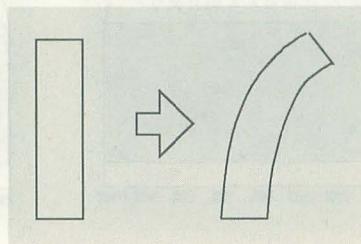
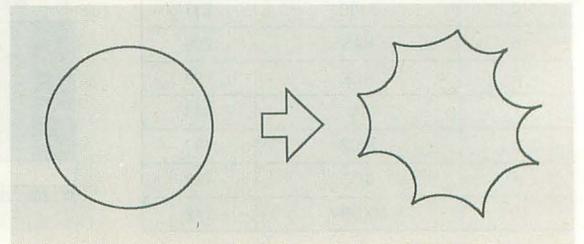



図3 大胆な軟体変形

種類があります。これらのツールは，恐竜の例のように，使い方によってはかなり面白い表現ができるにもかかわらず，使い勝手が悪いので，ついつい敬遠されがちです。そこで，これらのツールをいろいろ使ってみましょう。今月はEXPOINT.Xです。

EXPOINT.Xの基礎

EXPOINT.Xは，形状データの中の1つの頂点を摘んで引っ張るように変形させるツールです。ちょっと賢いのは，摘んだ頂点だけでなく，その付近の頂点も連動させたり，その連動の仕方もいろいろ指定できます。

しかし，まず使う前にいろいろと覚悟が必要です。1つ，非常に使いにくいこと。2つ，いろいろバグがある(と思われる)こと。そして3つ，理解しかねる仕様であること。なにしろプログラマ自身が，生まれて初めて作ったプログラムとはいえ，なぜあんな仕様にしたのかまったくわけがわからないといっているぐらいですから，使いこなすには相当の忍耐力が必要です。

たとえ“これは面白そうだから，次の作品制作のときに利用しよう”と思っても，作品制作に入ってからこのツールを初めて触るようでは，確実につまづいてしまうでしょう。作品制作に入る前に，あらかじめ十分練習しておくべきです。ということで，この記事をきっかけに触ってもらえれば幸いです。

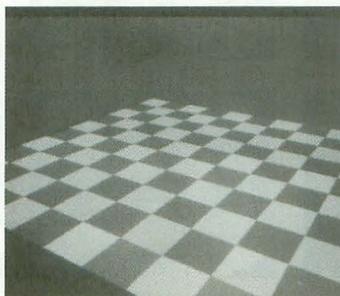
まず，写真Aが元の図形です。原点を中心に＋－500の平面ですので，ちょうど1マスが100になっています。この原点を摘んで，ちょいと上に動かしてみましょう。

どの点をどこに動かすかとか，周りの点の連動の仕方は，座標ファイルで指定します。この座標ファイルはエディタで書くしかありません。

B.TXT(リスト1)は原点をY方向に300もち上げるための座標ファイルです。

最初の1行は，(0，0，0)の点を(0，0，300)に移動するという意味だと想像できますが，あとはさっぱりわからないでしょう。これらは非常に難しく，かつ省略もできないので，あとでゆっくり解説します。とりあえず，実行してみましょう。

EXPOINT A /oB /pB.TXT

A.SUFという入力形状を，B.TXTという座標ファイルに従って変形させたあと，B.SUFという名前で出力しています。ここで注意点です。バグか仕様か不明ですが，B.SUFの中のオブジェクト名は，「obj suf A」のままになります。オブジェクト名を「B」にしたいときは，一度エディタで書き換える必要があります。また，当然ですが，「/p」オプションは省略できません。

上記のB.TXTで変形させた結果が写真Bです。

引っ張るのはどの方向でもかまいません。C.TXT(リスト2)のように原点を(300，400，300)に引っ張ってみると，写真Cのようになります。

アニメーションさせる

写真B，Cを御覧になって，皆さんはこのEXPOINT.Xをどのようなカットに応用できそうだと考えましたか。なにか伸縮性のある布を引っ張ったところ……そのまんまですね。トランポリン……もうひとひねりほしいな。ど根性ガエルが暴れているところ……歳がばれそう。でもテレビの画面の2Dのキャラクターが3Dの世界に飛び出そうとするカットなんかは面白そうですね。

どっちみち，これらのカットをアニメーションさせるためには，少しずつ引っ張る方向や量を変えた画像を連続して生成する必要があります。EXPOINT.Xを，RENCON.Xと一緒に使うことで，こうしたアニメーションが実現できます。

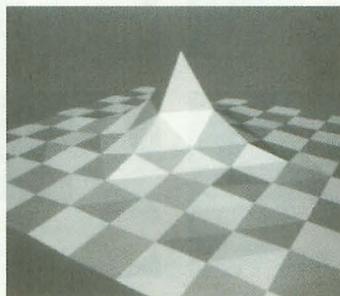
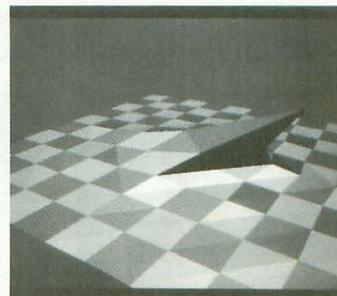
でもその説明をする前に，皆さんRENCON.Xのほうはちゃんと使えるのでしょうか？　RENCON.Xは，REND.Xで1フレーム作画させては終了し，ほかのツールやコマンドを実行し，またREND.Xで次の1フレームを作画させるというバッチファイルのようなツールです。この“ほかのツールを実行する”部分で，EXPOINT.Xを実行してやればよいのです。RENCON.Xを実行するためには，どのツールをどのように実行するかを指定す

リスト1　座標ファイル B.TXT

```
{
        expoint  ( 0 0 0    0 0 300 )
        limit    ( 500 500 500 )
        maxlevel ( 12 )
        type     ( hyperbola )
        rate     ( 50 )
}
```

リスト2　座標ファイル C.TXT

```
{
        expoint  ( 0 0 0    300 400 300 )
        limit    ( 500 500 500 )
        maxlevel ( 12 )
        type     ( hyperbola )
        rate     ( 50 )
}
```

写真A　市松模様の元画像

写真B　原点を(0，0，300)にもち上げる

写真C　原点を(300，400，300)にもち上げる

るコマンドファイルをエディタで書かなければいけません。C.TXTの変形を10フレームで行うコマンドファイルの例をD.CMD(リスト3)に紹介します。

ほかの変形ツールでも RENCON.X は必ず使いますので，D.CMDを少し解説しておきましょう。

まず1行目は，1フレーム目から10フレーム目まで作画させては，ほかのツールを使う動作を繰り返すことを意味します。以下「fno」と記述された部分にフレーム数が入ります。

2行目の「rendargs =」は，REND.Xを実行するときのファイルやオプションの指定をしています。記述の仕方はほぼ同じです。いちばん最後の「/sfno:fno」は，先ほど述べたように「fno」のところにフレームナンバーが入るので，たとえば3フレーム目を作画するときは，「/s3:3」になり，3フレーム目から作画をはじめて，3フレーム目で作画を終わる，つまり3フレーム目だけ作画して終了するという意味になります。ですから，RENCON.X のコマンドファイルには，この記述が必ず必要になるわけです。

3行目にEXPOINT.Xの実行方法が記述されています。RENCON.Xの機能としては，この部分にEXPOINT.X以外を記述することも，また複数行にわたっていろんなツールやコマンドを記述することもできます。

この行はA.SUFを入力して座標ファイルC.TXTに従って変形させてD.SUFを出力するという意味です。よく見るとこのD.SUFが2行目の「rendargs =」で使用されている点がポイントです。これで，1フレームごとに生成するD.SUFを読み込んで作画するため，少しずつ形状が変化するアニメーションになるのです。

そして「/f」が EXPOINT.X でアニメーションさせるときに必要なオプションです。たとえば「/f1,10,3」だと，“1フレーム目から10フレーム目にかけて変形するときの3フレーム目の状態”を出力してくれます。ですから「/f1,10,fno」とすることで，各フレーム違った形状がD.SUFとして出力されるわけです。

以上のようなコマンドファイルが準備できれば，

RENCON D.CMD

で実行です。できたアニメーションが，写真Dです。結局，エディタで新たに作らないといけないファイルは，座標ファイルC.TXTの6行とコマンドファイルD.CMDの3行だけです。思ったほどの手間ではないでしょう。

リスト3　コマンドファイルD.CMD

```
#frame( fno, 1, 10 )
rendargs = /a2 D.SUF A.ATR D.FSC /s$fno$:$fno$
EXPOINT A /oD /pC.TXT /f1,10,$fno$
```

変形タイプ

EXPOINT.Xの基本的な使い方がわかってきたので，座標ファイルをもう少し詳しく解説しましょう。C.TXTの4行目「type(hyperbola)」を書き換えることで，変形のタイプが変わります。

>比例

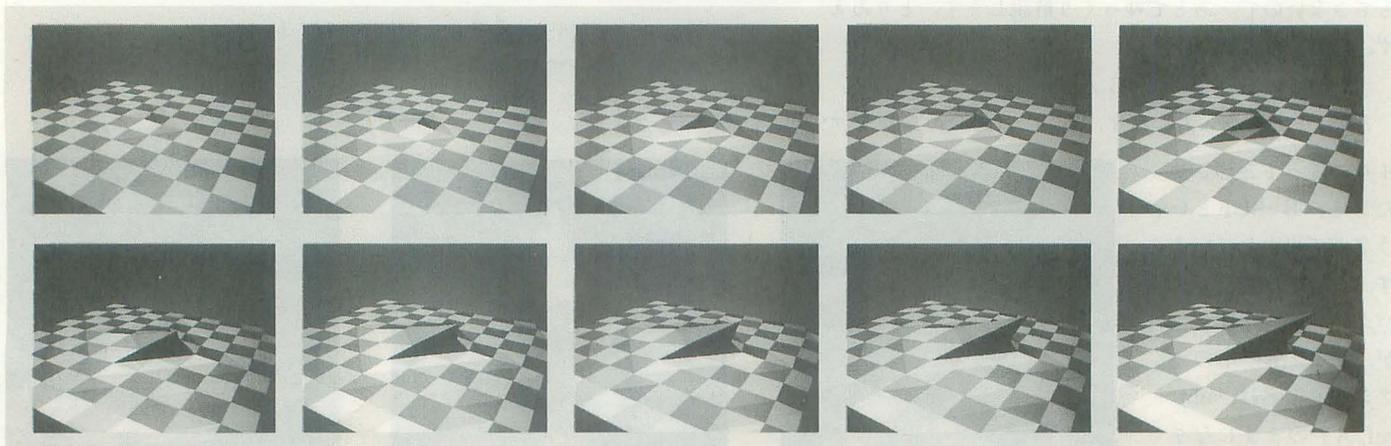
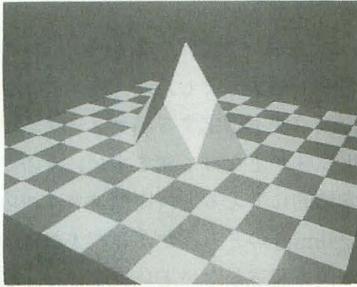
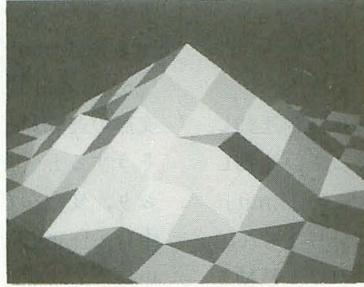
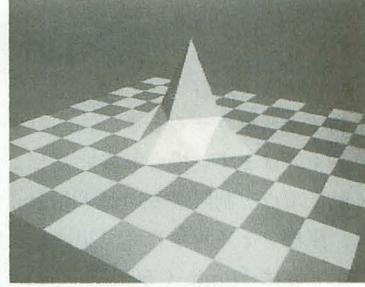
引っ張って膨らんだ部分が直線的になります。ちょうどポピュラスの山ができるような感じです。座標ファイルE1.TXT(リスト4)で変形させたのが写真E1です。

この場合，リスト1の4行目の「type」を「hyperbola」から「proportion」にしているだけです。次に5行目の「rate(50)」を「rate(20)」にしてみます。すると写真E2のようになります。また「rate(70)」にすると写真E3のようになります。つまりこの「rate」は，勾配を与えているのです。

「rate」は0～100の数値で，値が大きくなれば急斜面になります。通常は20～70ぐらいでしょう。

>コサインカーブ

同様に「type」を「coscurve」にするとコサインカーブ状に膨れます。座標ファイルF1.TXT(リスト5)によってできたのが写真F1です。

しかしながら，この膨らみ方(急勾配かゆるやかか)を指定するのは「rate」ではありません。「rate」はどのような値を入れてもまったく無視されます。ではどうするかといえば，「maxlevel」で指定します。このあたりが，

写真D　D.CMDを使ってC.TXTの変形を行ったアニメーション

写真E１　E1.TXTを利用して変形

写真E２　E１の勾配をゆるやかにした

写真E３　写真E１の勾配をきつくした

リスト4　座標ファイル E1.TXT

```
{       expoint  ( 0 0 0   0 0 300 )
        limit    ( 500 500 500 )
        maxlevel ( 12 )
        type     ( proportion )
        rate     ( 50 )
}
```

作者自身理解できないという仕様なのです。

「maxlevel」とは，摘んで移動する頂点と，その頂点につながっている頂点の中で最も近い頂点との距離を基本距離として，その何倍の距離まで連動の対象とするかを指定するパラメータです。たとえば，先ほどから利用している写真Aの市松模様の場合，摘んでいる点は原点で，そこからもっとも近い頂点までの距離は100です。そのとき「maxlevel」を「5」と指定してやると，原点から500以内の距離にある頂点が連動の対象となるわけです。

そこで，リスト5の「maxlevel」を「3」に変更すると，写真F２のように3マス目(300)以内の頂点が連動するようになります。ですから，この「maxlevel」を小さくしてやると急勾配に，大きくするとなだらかになります。

＞2次関数

「type」を「parabola」にすると放物曲面状に膨れます。勾配は「rate」で与えます。写真G１が「rate(5)」，写真G２が「rate(25)」の場合です。このように「rate」を大きくすると勾配が急になります。

＞反比例

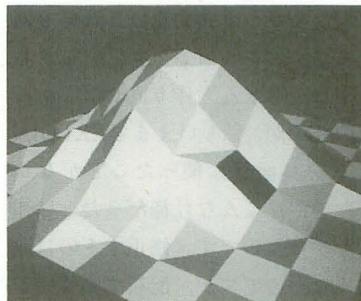
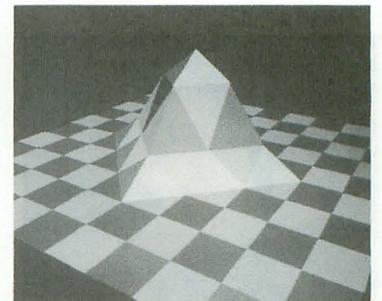
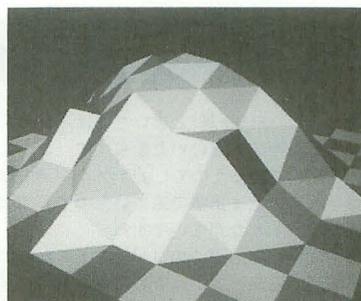
同様に「type」を「hyperbola」にすれば反比例になります。写真H１が「rate(50)」，写真H２が「rate(10)」です。参考にしてください。

＞なし

「type」を「nothing」にして，かつ「limit（０００）」とすると，連動が行われません(写真Ｉ)。ある特定の点だけを修正するときに便利です。

一般にEXPOINT.Xでは，点を移動させるとその周りの面を三角形に分割します。これは移動後の頂点が同一平面上に乗らなくなるためですが，REND.Xのほうは厳密にチェックしていませんので，少しぐらい同一平面上に乗らない頂点があってもちゃんと作画してくれます。ですから，EXPOINT.Xによる頂点の修正量が微小な場合，EXPOINT.X実行時に「/C」オプションをつけることで，無意味にポリゴンを分割して，ポリゴン数が増えるのを防ぐことができます。

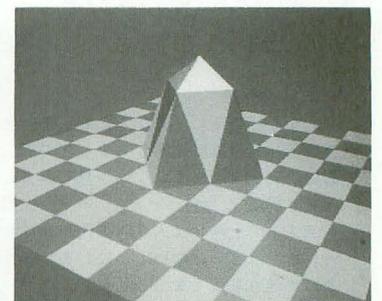
リスト5　座標ファイル F1.TXT

```
{       expoint  ( 0 0 0   0 0 300 )
        limit    ( 500 500 500 )
        maxlevel ( 5 )
        type     ( coscurve )
        rate     ( 1 )
}
```

写真F１　引っ張って膨らむ部分をコサインカーブに

写真F２　連動する点の範囲を変更

写真G１　膨らみ方が放物曲面になったもの

写真G２　写真G１を急勾配に変更

写真H１　膨らみ方を反比例に(rate＝50)

写真H２　勾配を与えるrateを10に変更

写真Ｉ　特定の点のみを引っ張る

変形範囲

物体を変形させるとき，物体のすべての頂点が連動して動くのはまずい場合も多いでしょう。そこで，連動して動く頂点を制限するのが「limit」です。この「limit」は，座標ファイル中に必ず記述しなければいけません。

たとえば「limit (500 500 500)」とすれば，移動させる点を中心にX，Y，Zの各軸方向に＋－500の範囲(直方体領域)の頂点だけが連動の対象となり，それ以外の点には影響を与えません。

実例として，リスト5の「limit」を小さくしてみましょう。「limit (200 200 200)」とすると，結果は写真Jのように，＋－200の範囲は従来どおり変形したものの，「limit」の範囲外ではまったく変化がありません。写真F1と比べてみてください。

謎のmaxlevel

しつこいようですが，以下の仕様については，作者自身理解できないほどですから，私に“そんなバカなわけないだろう”と文句をいわれても困ります。バージョン2.Zを配布するときに，ちゃんと作り直せばよかったのですが，私もいままでEXPOINT.Xを真面目に使ったことがなかったので，こんな仕様だったとは知らなかったのです。修正するのは難しくないのですが，配布する手段がありません。とりあえず，諦めていまのバージョンを使ってください。

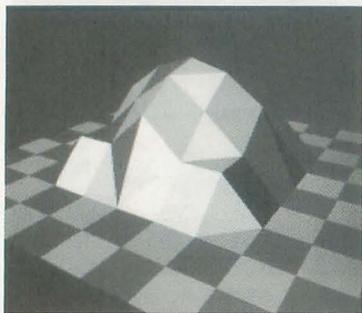
写真Ｊ　limitで連動の範囲を狭くする

まず，「maxlevel」の意味ですが，変形タイプの「coscurve」のところで少し解説しましたが，基本的には変形の連動がどこまで及ぶかを指定するパラメータです。そうすると「limit」と意味が重なって変ですが，そのとおり，変なんです。常識的に考えると，座標ファイルの中に，「maxlevel」と「limit」のどちらかが存在すればよいはずですが，なぜか両方記述しないといけません。結局，連動する範囲は，「maxlevel」で指定された範囲かつ「limit」の範囲となります。

「limit」と異なる点は，「limit」が連動の対象となる領域を具体的な数値で表すのに対して，「maxlevel」は隣り合う最も遠い頂点との距離(基本距離)の何倍までというように指定します。したがって，同じ「maxlevel(10)」でも，隣り合う頂点の位置によって，範囲はまったく違ってきます。また，「limit」が直方体領域を指定するのに対して，「maxlevel」は移動する頂点からの距離，つまり球体領域を指定することになります。

連動量の計算

さらに変なのは連動量の計算方法です。この計算では，上記の「maxlevel」の基本距離が重要になり，この基本距離単位の精度しかありません。つまり，基本距離の2倍以上，3倍以下の範囲にある頂点はすべて連動量が同じになるということです。

たとえば，放物曲面「type(parabola)」の場合を考えてみます。本来ならば，図4のように，頂点をもち上げて放物曲面を生成する場合，理想的には点線のようになり，各頂点は△のマークの位置に連動するはずです。そ

コンテストアンケート

ちょうど手元にCGAコンテスト東京会場でのアンケートの集計結果が来ましたので，なにか面白いデータが出ていないか見てみましょう。

まず男女比。「女性：1.4%」……。次回からコンテストの上映会は入場料の代わりに女性を同伴しなければいけないことにしましょうか。ほとんど誰も来なくなったりして。

年齢は，19～26歳が多いですね。使用機種はX680x0が68.7%とさすが。まぁ，約半数の方がOh!Xでコンテストの情報を得たといっているので当然ですか。使用機種の2位はPC-98シリーズおよびその互換機で25.2%，3位はMacintoshで12.9%となっています。CGAコンテストはX680x0用のコンテストではありませんから，ほかのユーザーももっと見に来てほしいですね。

上映会の感想としては，「(非常に)面白い」が90.5%，「普通」が9.5%，「あまり面白くない」「ぜんぜん面白くない」が0%。なかなか好評だったようですね。また，「来年もまた来たいか」で「来る」がほぼ全員というのもすごいですね。〝ほぼ全員〟と書いたのは，「たぶん来ない」の人が1人だけいたからです。この人はなにかご不満だったのでしょうか？「今日は北海道からたまたま出張で来ていたので」。なるほど，それはしかたがないですね。

でも，このコンテストのためにかなり遠方から来たという方も結構いらっしゃいます。仙台，新潟，岐阜，名古屋，……京都，大阪，岡山？そんなん大阪会場のほうが近いやん。

上映作品への感想としては，「年々レベルが上がっている」が53.7%，「凄い手間をかけている」が35.4%，「非常に面白い」が32.0%。その半面「ストーリーが弱い」も34.7%と多くなっています。

面白いのは「好きな作品」と「嫌いな作品」です。人気があったのは，1位「二進数の世界」，2位「電神ギガダイン」，3位「A DRAGONFLY」，4位「GENESIS」，5位「超からくり刑事」。お笑い路線に人気があるようですね。ちなみに東京会場では「二進数の世界」，大阪会場では「超からくり刑事」が人気投票による会場審査員特別賞を受賞し，IMAGICAテクノシステムより提供していただいた，XL/Imageが贈られました。東京と大阪ではお笑いも違うようで……。

逆に「嫌いな作品」はストーリーのない作品が多くなっています。そんな中でワースト3に「GENESIS」が入っています。いいですねぇ。好き嫌いがはっきり分かれる作品だと予想したとおりです。やはり佳作の作品はこうでなくっちゃ(詳しくは解説本をごらんください)。

「審査結果について」は「まぁ順当なところ」が86.3%と大方納得されたようです。そのほか，上映会の長さ，展示内容，座談会や質疑応答にいたるまで，運営するうえで貴重なご意見をいただきました。これを参考に，今後もより楽しい上映会にしていきたいと思いますので，来年もぜひいらしてください。

なお，4月号についていた振り込み用紙によるビデオの申し込みは，4月30日までとなっていましたが，これはOh!Xの読者だけ先に発送するためで，一般からの申し込みは5月31日まで受け付けています。まだ間に合いますので，5月号を見て，申し込んでください。

の結果，図5のような曲面になります。

しかし，移動量は基本距離単位で同じになりますので，実際には図6のように移動します。その結果，それらの頂点を結んでできる曲面は図7のようになります。つまり，滑らかな曲面にならずに，ガクガグした形になります。なぜ，このような仕様になっているかは，まったく意味不明です。……あっ，石を投げないでください。

いままでサンプルにしていた写真の多くがガクガクしていたのは，まさしくこれが原因です。比例のモデルなんかでも，理想的には円錐状に膨らむはずですが，角錐になってしまっています。

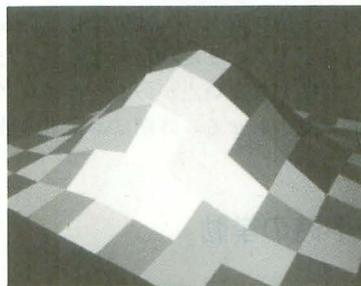
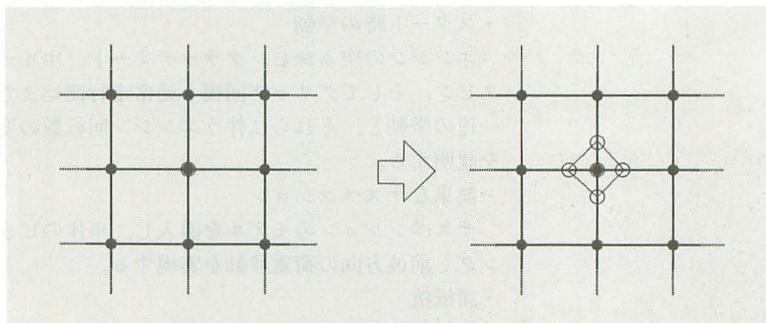
しかしながら，原因がわかれば対処方法を検討することができます。たとえば，基本距離が小さくなれば，精度が高くなるはずです。そこで市松模様の床でも，図8のように，引っ張る頂点に隣り合う頂点をみんな近づけるようにモデリングしてやると，同じコサインカーブでも写真Kのように滑らかになります。写真F1と比較してみてください。さらに，SHADE.Xで法線ベクトルを平均化すれば，写真Lのような美しい曲面が得られます。

図4　理想的な放物曲面

図5　理想的な頂点の連動から生成される曲面

図6　実際の連動量

図7　実際に生成される曲面

図8　基本距離を小さくするモデリング

写真K　モデリングを図8のように変えてみる

写真L　SHADE.Xで法線ベクトルを平均化

より高度な使い方

より高度な使い方といっても，以下の2点はマニュアルにもちゃんと載っているので，簡単に解説します。

まず，座標ファイルには，同時に複数(最大16)の頂点を移動させることができます。それぞれの変形は，変形タイプも変形範囲も違っていてかまいません。ただ，注意しないといけないのは，たとえば原点を(0，0，300)に動かしたつもりでも，ほかの点を移動させた連動で，最終的に(0，0，300)には移動しないことがあります。

もうひとつは，アニメーションに便利な機能です。「expoint(0 0 0　0 0 300)」は通常の移動の指定ですが，N1フレームではこの位置，N2フレームでこの位置というように，フレーム数とその位置を指定する方法があります。「expoint」の代わりに「div(N1 X Y Z　N2 X' Y' Z'　……)」と記述します。フレームソースの「div」と似ているのですが，パラメータの並べ方が違っているところがなんとも EXPOINT.X ですので，ご注意ください。

おわりに

私自身，今回初めて本格的にEXPOINT.Xを使ってみた感想としては，連動量の計算の誤差の問題が大きく，実用性はほとんどないという結論になってしまいました。

しかし、まったく使いものにならないわけではありません。事実「DRIVIN' WOMAN」の笑顔の変形はEXPOINT.Xで行われています。ただ，この場合，連動は使っていないと思われます。また，連動するとガタガタになる点を利用して，地形を作成するのもよいかもしれません。……ちょっと苦しいですか。少なくともあのマニュアルだけではわからない点も多いので，詳しく解説させていただきました。

次回は，変形ツールの第2弾BOXTRANS.Xです。実は作者がEXPOINT.Xと同じ人です。しかし，その後いくつものプログラムを開発したあとのツールなので，仕様はずっとまともです。実用性も高くなっています。ということで，お許しください。

SIDE A

ゼロヨンゲーム完結……かも(前編)

Tan Akihiko 丹 明彦

簡単な運動モデルを確認するためのゼロヨンモデル
いよいよ完成にむけて最終段階の詰めに入る
しかし，最終目標に向けてまだまだやるべきことは山積みだ

さて今回はゼロヨンゲーム，つまり自動車の直線加速運動のシミュレーションに必要な要素のうち，残ったものを入れてしまうことにする。

・スタート時の挙動

エンジンの空ふかし，クラッチミート，ホイールスピン，そしてグリップ回復と通常走行開始までの一連の挙動と，それらに伴うエンジン回転数の変動を説明する。

・簡単なサスペンション

サスペンションのモデルを導入し，車体のピッチングと前後方向の荷重移動を実現する。

・諸抵抗

空気抵抗と路面からの抵抗を導入する。

現在のところ，一応プログラムは動作している。ただ，ゲーム性という点でまだ物足りなく，また，1カ所論理的に説明できない点があるためプログラムリストの掲載は見合わせる。詳細は本記事の最後をご覧いただきたい。

スタート時の挙動

エンジンの回転数はタコメーターで知ることができる。自動車の挙動の多くがエンジン回転数とそれに対応する出力で決まることを考えると，タコメーターの針の動き方をきちんと表現できるモデルはとても大切である。まず現象面から見てみよう。

スタート前にアクセルをふかすと，タコメーターの針は軽々と上下動する。ちなみに，開発中のゼロヨンゲームでは，車ゲームのお約束として，スタート前はクラッチが強制的に切られている。なお，停止中かつブレーキを踏んでいる間もクラッチを切っているとみなして，空ふかしが可能になっている。

スタートの瞬間クラッチがつながり，駆動輪が回転を始める。このときエンジンの回転数が高いと，タイヤと路面の間のグリップは失われ，タイヤは高速で空転を始める。これがホイールスピンである。タコメーターの針はクラッチミート時の回転数をほぼ維持する(タイヤを回転させ始める仕事のために多少は食われる)。

逆にクラッチミート時にエンジンの回転数が低いと，タイヤはグリップしたまま，車は通常の加速(エンジンの出力がタイヤを通じてダイレクトに路面に伝わる)を始める。タコメーターの針はいったん0近くまで落ち込み，それから加速するにつれて上昇を始める。

ホイールスピン中は車はあまり進まないが，タイヤは空転しながらでも少しずつ路面を蹴っているので，車速は上昇していく。そのぶんの仕事でエンジンの回転数は食われるが，アクセルを開け続けていれば損失分は補填されるのでさらにエンジンの回転数が上がる場合もある。アクセルをゆるめればエンジンの回転数は落ち込む。タコメーターの針もそのとおりの動きをする。

ホイールスピンしながらの加速によって車速が十分に上がるか，またはアクセルをゆるめることによってタイヤはグリップを取り戻す。つまりエンジン回転数，すなわち駆動輪の回転数が落ち込むことによって，駆動輪の接地点と路面の速度差が少なくなった瞬間だ。その瞬間エンジン回転数は速度に応じた値に落ち込む。このあとは通常の加速に入る。タコメーターの針もそのように動く。

これらの一連の挙動を計算機上で実現するために，図1のようなモデルを立ててみた。エンジンの発生するトルクはまずフライホイール(はずみ車)を回すのに使われ，そこからクラッチとトランスミッションを通じて駆動輪を回すのに使われ，さらに路面との接触点を通じて車体を前進させるのに使われる。しばらく前にも似たような図を載せたが，エンジンの置き方が変だと御指摘をいただいた。今回の図にも変なところがあるだろうが，前回同様に概念図ということで御容赦いただきたい。

クラッチを切った状態では，フライホイールだけ

が回転する(図2)。フライホイールが重いか軽いかで，空ふかしのときのエンジン回転数の上下する速度が決まる。

クラッチはつながっているがタイヤが路面をグリップしていない状態では，フライホイールから駆動輪までが回転する(図3)。当然，回転部分はフライホイールのみよりも重いので，アクセル操作に対するエンジン回転数の反応は鈍くなる。

そしてタイヤがグリップした状態では，さらに車体も動く範囲に含まれる(図4)。車体ははるかに重いので，アクセル操作に対するレスポンスはこの状態が最も鈍い。

これら3つの状態は明確に異なるため，状態を切り換える際に滑らかな変化をさせられない。それまでの回転部分がもっていた運動の状態を，新たな回転部分が引き継ぐ様子を実現するために，今回は「角運動量の保存則」を用いることにした。角運動量とは以前も紹介した慣性モーメントに角速度を乗じた量である。高校の物理では，運動する質点の衝突における運動量保存則を学んだが，角運動量保存則はちょうどこの回転運動版と考えればよい。この法則を導入すると，複数の回転運動をする物体が一体になった瞬間の，その新たな物体の回転運動を決めることができる。そしてそれはクラッチミートやタイヤのグリップ回復の瞬間の挙動にも適用できるはずである。

なお，今回は半クラッチ，すなわちクラッチが完全には接続せずに滑り，エンジンの出力の一部しか伝達されない場合の挙動は導入していない。タイヤのグリップと状況が似ているため，特に積極的に入れる必要性を感じなかったし，プレイヤーが混乱するのを防ぎたかったからである。画面と音のみから得られるインフォメーションでは半クラッチの表現は難しいかもしれない。

サスペンション

サスペンションといっても，リンクで構成された構造物のシミュレーションまで行うわけではない。スプリングとダンパーを対にした単純なユニットで

図1　乗用車の駆動システムのモデル(概念図)

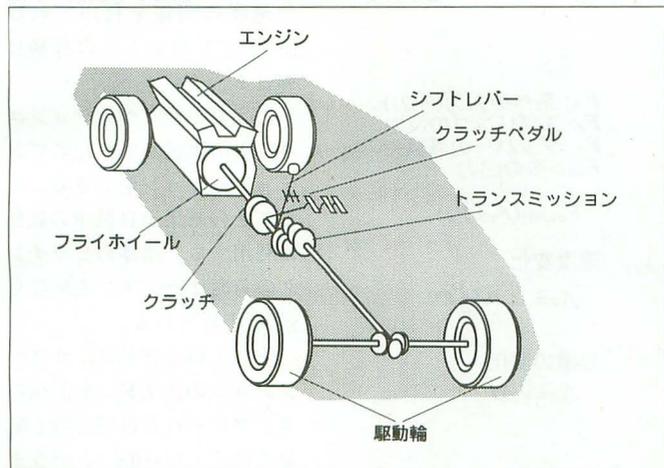



図2　クラッチを切っている場合

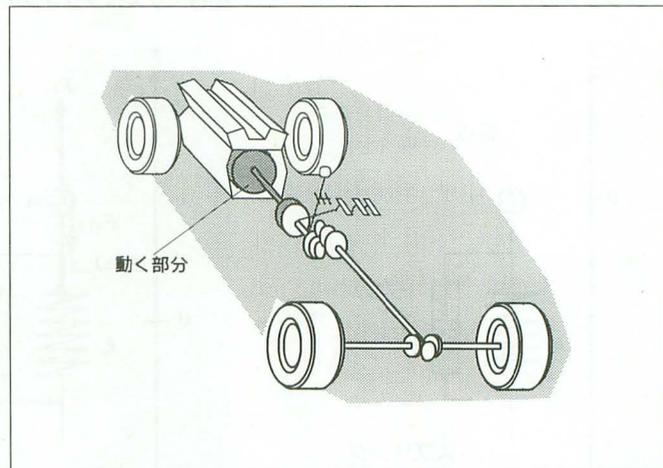



図3　ホイールスピンしている場合

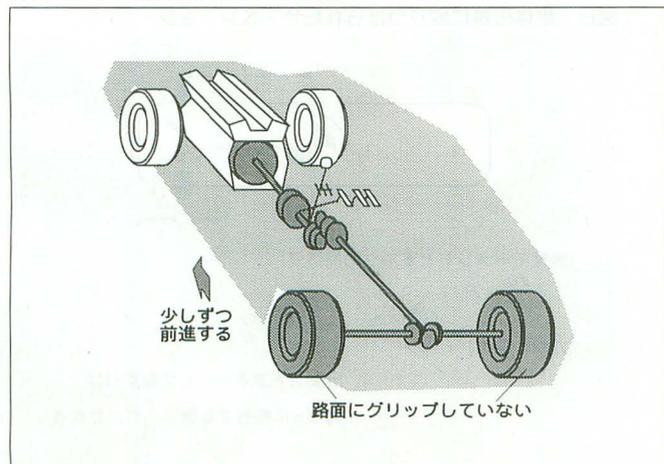
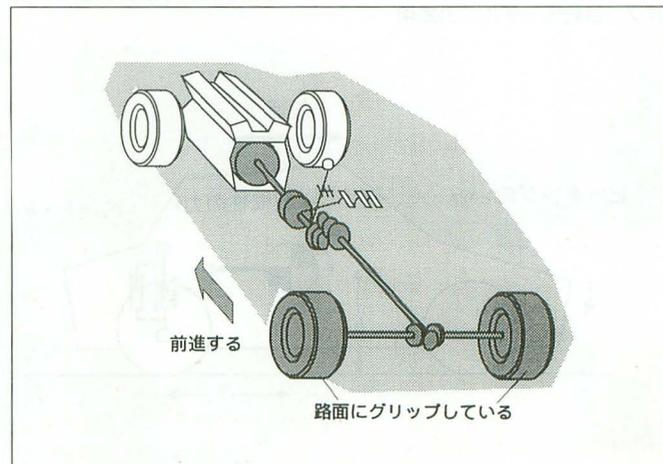



図4　通常走行している場合

ある(図5)。

スプリングは変位(伸び縮みした量)xに比例して変位とは逆方向に力を発生する。そのばね定数をk_sとする。ダンパーは速度vに比例して速度とは逆方向に力を発生する。その係数をk_dとする。

このサスペンションユニットの先端に質量mの質点を取りつけて外力F_{in}を与えた場合の動作を図6に示す。このような性質をもつサスペンションの挙動は,

・外力に従って最初は縮む(伸びる)が,その後逆に伸び(縮み),以後伸縮を繰り返す。これはスプリングの性質による。

・質点に速度がついている場合はその速度が鈍り,ある時間内にニュートラルの位置に戻る。これはダンパーの性質による。

というものになる。ばね定数とダンパーの係数を変えるとさまざまな特性を備えたサスペンションが作れる。

さて,このサスペンションユニットを自動車に組み込むわけだが,今回扱うのは水平面上の直線加速運動ということもあるので,最も単純な実装にする(図7)。後輪車軸中央に仮想的なサスペンションを取りつけたと考えるのだ。

サスペンションが関係するのは,車体のピッチング(縦揺れ)と前後方向の荷重移動である。さきほどのサスペンションユニットをそのまま用いたいので,自動車の状態変数をサスペンションの仕様に合うように変形することにしよう。

サスペンションの入力には,車体の加減速に伴って生じる慣性の力を用いる。加減速は水平方向だが,車体は剛体なので車体を回転させようとする力が働き(これが最終的にピッチングとなって現れる),サスペンションにも垂直方向の入力が生じる。慣性の力をF,重心点までの高さをh,重心からサスペンションの取りつけ位置までの水平距離をlとした場合,サスペンションの入力は$F_{in}=F\times h/l$である。

剛体である車体の運動を,質点の運動を記述していたさきほどのサスペンションに適用するには,慣性モーメントIを質量mに相当する量に翻訳する必要がある(図8)。トルク(重心と力点の距離に力をかけた量)と慣性モーメントと角速度の関係を利用すれば$m=I/l^2$であることは容易に導ける。

こうしてサスペンションユニットの実装ができたのでシミュレーションができる。そこでその結果を自動車の状態に利用する。車体のピッチングの角度はラジアン表記ならx/lで求められる。

そして荷重移動は,サスペンションの出力F_{out}を前後のタイヤにそれぞれ逆符号で配分すればよい(図9)。いうまでもなく,加速中は後輪の荷

図5 スプリングとダンパーによるサスペンションのモデル

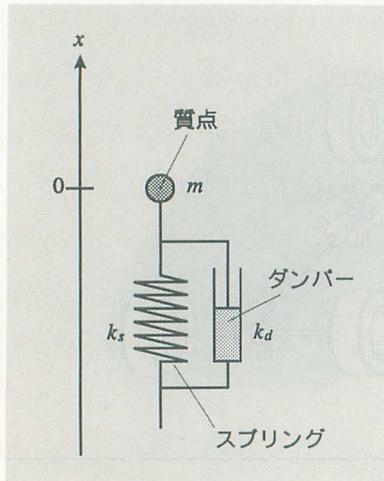



図6 サスペンションの動作

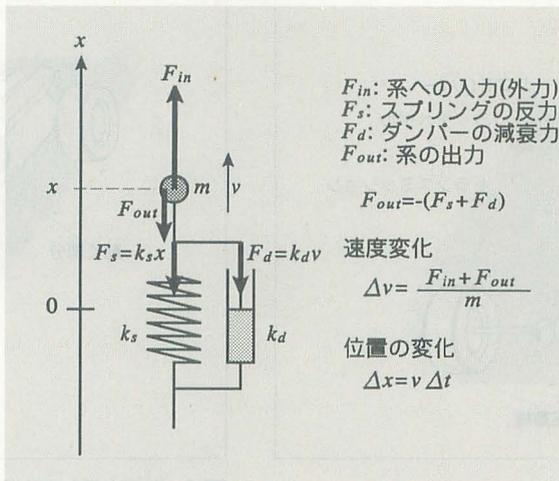



図7 自動車モデルへの応用

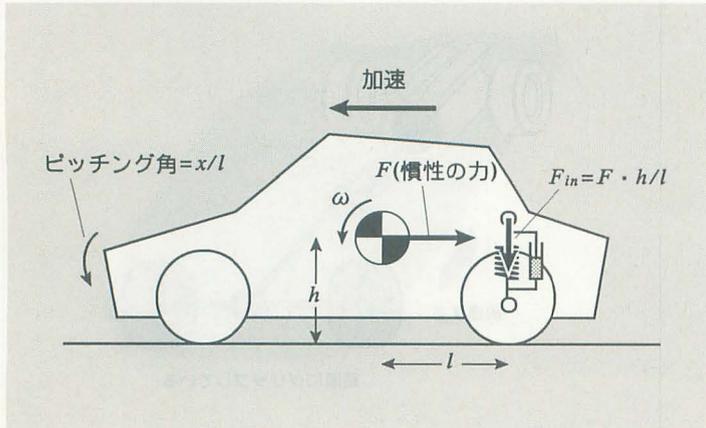
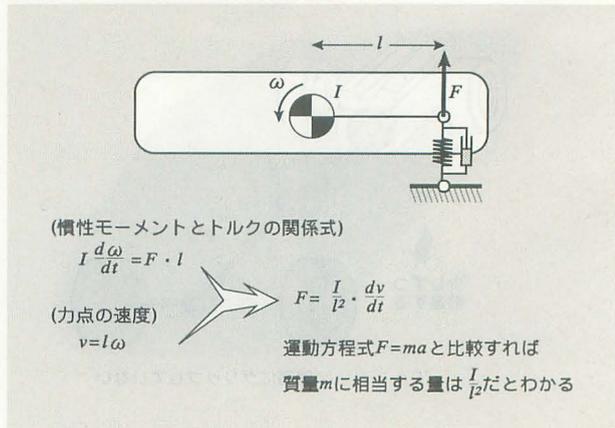



図8 剛体の端に取りつけられたサスペンション

重が大きくなる。これは前輪駆動車が急加速を不得意とする原因にもなっている。事実，このゼロヨンゲームでも，前輪駆動に設定するとタイムはあからさまに落ちてしまう。ただ，この傾向は，初期重量配分を変えることによってある程度回避できる。たとえばフロントの重量配分を大きくすればよい。

空気抵抗と路面抵抗

ここは楽である。

空気抵抗は車速vと車体の空力特性(いわゆるCd値と呼ばれる抗力係数と前方投影面積S)，それにその場の空気の密度ρによって決まる。

$$空気抵抗=(\rho\cdot S\cdot Cd\cdot v^2)/2$$

空気密度は定数としても当面はかまわないが，高速で走行する車の後ろでは空気の密度を下げれば，非常に簡単にスリップストリーム走行をモデル化できることは知っておくと便利だろう。

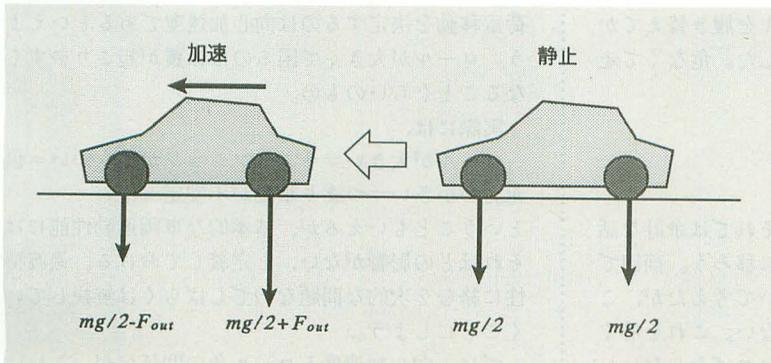
そして，路面抵抗は基本的に車速vによって決まる。

$$路面抵抗=\mu(v)mg$$

路面抵抗係数μの値は，速度がそれほど高くない間は無視できるほど小さいが，時速140km/hを超えるあたりからスタンディングウェーブ現象が起こるため急激に増大する。路面抵抗のゼロヨンゲームへの実装にあたっては，車速vを横軸に取った折れ線グラフを用いた（図10）。

解決できなかった部分

本来ならば今回でゼロヨンゲームは完成するところだったのだが，ひとつ問題が残ったために完成を宣言することができない。明確にモデルを立てることができない現象が残ってしまったのだ。

その問題とは具体的にいえば「ホイールスピン時の加速」である。どの程度加速して，どのくらいエンジン回転数が落ち込むのかを定量的に議論した資料が見当たらない。エンジンのフライホイールの軽さに対する情報がないことも事態をいっそう複雑にしている。

現時点ではエンジン回転数の落ち込みに対して意味不明の係数をかけるという，非常に場当たり的な方法でこの問題を回避している。だから，普通にプログラムを実行してもインチキしていることはばれない。しかしこの場で数式を使って説明するというような恥知らずな真似もできない。このあたりが，曲がりなりにも完結した理論体系の中でシミュレーションモデルを構築しなくてはならない本連載の難しいところである。

次回までにはこの問題を解決したい。できれば，クラッチミートとアクセルコントロールのテクニックがロケットスタートに結びつくところまでは実現したいと考えている。

次回予告というか今後の展開

ゼロヨンゲームもようやく完成にこぎつけそうな雰囲気だ。そこから先は思案中なのだが選択肢は，

1)　その時点でいったん区切りをつける。ゲームとしての体裁を整え，16ビット機向けに最適化も施し，完全な形で発表する

2)　かまわず先に進む。さらに複雑な自動車の運動を組み込んで，引き続き浮動小数点演算の嵐で突き進む

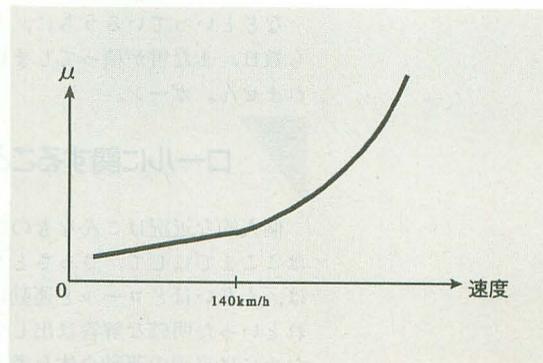
の2つである。1)はそれなりに手間暇がかかるし本連載の最終目的にとっては回り道でしかないのだが，もう長いこと読者からは再現不能な内容を続けてしまっているといわれているのが気になる。まあ安易な折衷案として，

3)　X68030＋68882専用という但し書きつきで発表する

という選択肢も存在することは存在するが，これでは発表する意味はほとんどないか。

今回はゴールデンウィーク進行に敗れたので，この遅れをゴールデンウィークに取り戻すつもりだ。というところでまた次回。

図9　荷重移動

図10　路面抵抗

SIDE B

車の基本運動と力学

Yokouchi Takeshi 横内 威至

今回も車の基本運動を力学的に見ていく
これで，車におけるだいたいの運動の関係が把握できるだろう
シミュレータに実装するためには，きっちりと理解しておきたい

雪が路面から消えました。タイヤを買いました。205-40の17インチとかいう，3インチアップの非人道的なサイズです。本当は全然スポーティでないけど，なんとなくスポーティそうなボディに輝く巨大な銀輪。リッジレーサーの車っぽいです。車高は高いけれど。みんなはバカだというけど，嬉しい。でも実際届いてみると不安です。ゴムの部分がほとんどありません。ちょっとした段差でも軽くホイールがブッ壊れそうな薄さです。乗り心地は極めて低下しました。ロードノイズも以前と比べるとかなり大きく，ステアも重く，乗ってて疲れるばかりです。性能はどうかというと，路面が怖くてスピードが出せないからチェック不可能です。後輪のグリップが強まってアンダー，というところだと思います。サスペンションもいずれ替える予定なので，そのときにやっとセッティングの効果を味わえるのでしょう。しょせん，車はカッコだけです。速さなんかはどうでもいいです，俺は。

とかいいつつ，速い車を操るのは好きです。やろうと思っていることと実情が矛盾していますが，それはいつものことです。勝負に負けて絶対に玉は弾かない，と心に誓ってもパチンコ勝負に出てしまうのはパチンカーとしての悲しい性というもの。皆さんもそうですよね。でもパチンコはやめましょう，大人なんだから。

などといっているうちに，タイヤを履き替えてから数日，また雪が積ってしまいました。危なくて走れません。ガーン。

ロールに関すること

個人的な近況はこんなものだ。それでは余計な話はここまでにして，さっさと本題に移ろう。前回では，くどいほどロールと運動について考えたが，これといった明確な解答は出していない。これを解くためには車両の運動全体を考えなければならない，という結論を出しておいた。このこと自体は間違いではないのだが，実はちょっとばかりウソがある。

今回は，このあたりのウソをなくすためにしっかりと理論を固めることから始めよう。まずロール自体の本当の影響を考えてみよう。ロールすることによって車体が傾き，荷重移動が起こる。荷重移動が起こることによって，グリップに変化が生ずる。これがロールによる車両運動への影響だ。

ロールが大きいとどうなるか。ちょっと考えただけでも，なんとなく荷重移動量が大きくなるし，重心の位置も激しく動いてしまって，運動の制御が不安定なものとなるのはわかるであろう。ロールが小さいと逆である。ドライバーとしては，ロールが小さいとコーナリングに対して安定感があるため，安心してコーナーを攻めることができる，というのはちょっと強引か。

では，同じ軌道，つまり同じ向心加速度のときにロールが大きい場合と小さい場合について比べてみよう。ロールが小さいほうが荷重移動が少なくて安定している，ということはない。

荷重移動量は，

$$\Delta W = M \cdot a \cdot h / T$$

であり，ロールによる影響は通常それほどのものではない。ロールによってh（重心高）が変化するが，これによって大きく変化するものではない。つまり，荷重移動を決定するのは向心加速度であるといえよう。ロールが大きくて困るのは転覆が起こりやすくなることぐらいのもの。

実際には，

ロールが大きい＝サスペンションが柔らかい＝復元力が小さい＝つまり挙動が不安定っぽい

ということもいえるが，基本的な車両運動性能にはそれほどの影響がない，と定義しておける。過渡特性に絡む2次的な問題なのでしばらくは無視していくことにしよう。

では，向心加速度とロール角の関係だが，これは

サスペンションのバネ係数によって変化する，といえる。車のメーカーでも，ある向心加速度のとき，普通のファミリーカーならば6度，スポーツカーならば4度，というような感じで決定しているらしい。つまり，これは適当なテーブルを用意してロール角を決定することで，ロールを再現することができることになる。

キャンバーなどの接地状況によるもの

これもロールによって状況が変化するものである。キャンバー変化はロール変化とともに変化するものであり，これもテーブルを作って変化を再現することが可能だ。車両運動を計算するときに重要なのは，運動に影響を与えるレベルの力を発するという点のみ。

さらにトー変化なんかもあるが，これは車両の運動には微小な影響しか及ぼさないのでとりあえず無視しておく。まあ，これも接地状況によって変化するものなので，同じように状況を調べてから影響を加えることで十分だろう。

運動への影響とは

いままで「運動に影響を与える」かどうかを考えて，そのうえでどのような手順で運動を計算するかを探ってきたわけだが，影響とはどのような形で与えられるかを考えなければならない。極めて当然のことなのだが，これは車両に力が働くことである。物理で扱っていた「力」のことである。

では，車は一般の運動方程式，

$$F=ma$$

で表すことが可能だろうか。残念ながら無理だ。なぜなら，これらの力が働く部分は重心とは別の点で，重心と一体になって運動する4点だからである。この複雑な運動は，いわゆる「剛体」の運動なのである。

そして大学の一般教養で教わるような単純な剛体のモデルとは掛け離れているものだろう。非常に残念ながら，教養過程の不真面目さが裏目に出たのか，私は剛体の力学にはかなり疎い。本当に役に立つのかどうかわからないが，これを理解していないのはかなり悔しい。ということで大学の物理の教科書を開いたがまったく理解できない。これから大学に通う方にいっておくが，どうせヒマなんだから授業はなんとなく理解しておくと，もしかしたら役に立つときが人生に1回ぐらいはあるかもしれない（人生なにがあるかわからないからな)。だから一応大学でもがんばろう。

さて，基本的には剛体力学の問題だが，そう簡単にはあてはまらない。よって，独自の解析を行う必要がある。いろいろな自動車力学の本を読んでも力と運動の関係は完璧ではない。というより，理論は完璧なのだろうが実際にシミュレートするときに重要で，簡単に解くことができる式を導いている本はいまのところ知らない。具体的には力とモーメント，そして車重との関係から一定時間後の姿勢，速度を計算するためには，ある程度の力技が必要となりそうなのである。

ということで，力がどのように働くか，そして運動は力でどのように変化するかを考えることにしよう。

タイヤの力学

車が曲がるのはなぜかというと，タイヤが力を発生させているからである。当たり前だけど，タイヤじゃなければ曲がらない。たとえばスキー板がタイヤの代わりについていたらどうか。いくらステアを切っても車は直進しかしないはず。物理的にどうなっているかはわからなくても，感覚的に理解できると思う。どこで差が生まれるのかというと，タイヤは路面をグリップすることができるがスキー板ではグリップできない点である（たぶん)。

では，ちょっと進んだ部分ではあるが，タイヤが路面をグリップしないとどうなるだろうか。というより，どうなるとグリップを失うだろうか。これも感覚的に，タイヤの回転が止まるときがそうであるのはわかると思う。急制動は危険，といわれるのはこのような理由からだ。タイヤは一定以上の力を発揮することができず，それ以上の力を加えるとロック（固定）してグリップを失い，単純な摩擦以外には力を発揮することができなくなる。摩擦は速度を失う方向に力を加えるだけなので，コーナリングをすることができなくなる。コーナリング中のブレーキングが危険といわれるのはこのためである。実際にはもうひとつの理由もあるが，ロックが危険であるということは同じである。

タイヤから発生する力を図1に挙げておく。どのような状況でこれらの力を発揮するか説明していこう。まず，タイヤの回転に対する抵抗が，転がり抵抗，つまりタイヤが回転しているときにそれを止めるような力を発揮する。制動力はブレーキなどによって，やはりタイヤの回転を止めようとする力である。そして次に，最も重要な横力，あるいはサイドフォースである。これはタイヤの進行方向とタイヤの向きが違うときに発揮される。また，この角度を横すべり角，スリップ角などと呼ぶ。これらの力はタイヤが速度をもたないときには働かない。あとはタイヤの回転を促進させる駆動力，タイヤが垂直で

ないときにかかるキャンバースラストがメインの力である。

また，運動とは関係のないセルフアライニングトルクもかかる。これはタイヤが発揮する力によって，逆にタイヤを進行方向に向けさせようとする力である。タイヤが力に負けず，車体を回転させているのはこの力にステアリングで抵抗しているためである。グリップの高いタイヤを履くとステアリングが重くなるのは，このセルフアライニングトルクがタイヤの発揮する力とともに大きくなるからだ。

これらの力のベクトルを合成したものがタイヤの発揮する力となる。特に進行方向と直角の成分をサイドフォースと呼び，これが車体の回転運動に結びつく直接の力となる。

サイドフォースが重要といったが，これはどのようなものであろうか。ステアリングを切れば車体はより小さい半径で回ることからも，スリップ角が大きいほどサイドフォースも強くなることがわかる。また，スリップ角とコーナリングフォースの関係は図2のようになり，ある程度の角度で限界となる。コーナーで焦って急ハンドルを切っても，車が思ったように曲がらないのは当然なのである。

キャンバー角とキャンバースラストの関係は図3。キャンバー角が大きいほどより大きなキャンバースラストを発揮する。なぜこのような力が発揮されるか説明することは面倒だが，とりあえず丸い円盤状の物体を斜めにして転がすのをイメージすれば理解できるだろう。ちなみにドリフト仕様の鬼キャン，キャンバー角を大きくと

図1 タイヤから発生する力

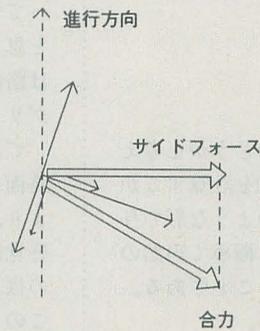
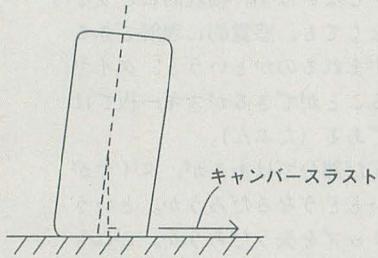
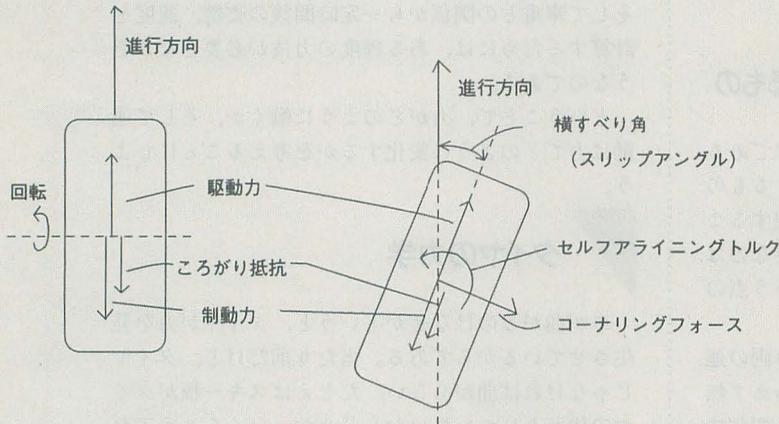



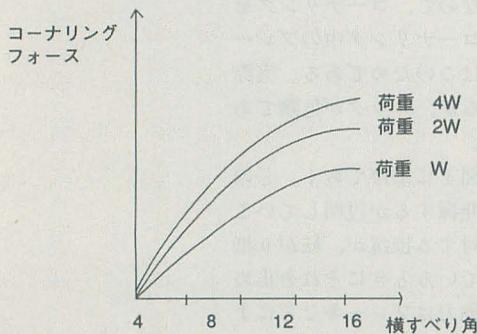
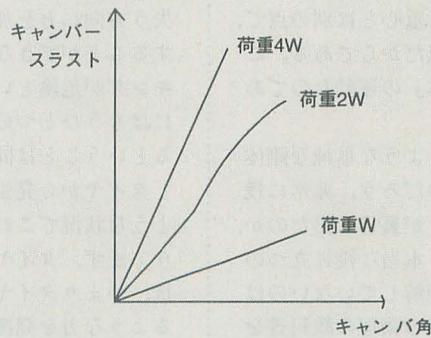
図2 スリップ角とコーナリングフォースの関係

荷重が大きいほどコーナリングフォースは増加する。
また，スリップアングル16°ぐらいで限界となる。

図3 キャンバー角とキャンバスラストの関係

キャンバー角に応じてほぼ直線的に増加する。
車の場合，キャンバースラストはコーナリングフォースの10%程度となる。バイクは見てのとおり90%は占めているであろう。あれほどまで強烈にキャンバーを傾けても限界ではなく，角度に比例して増加することがわかる。

って前輪をハの字にすることでキャンバースラストを稼ぐ意味はたぶんないと思う。いくら大きくしたところで，結局左右で打ち消し合う力となるので意味がないはずだ。むしろこれで前輪のトレッド（車輪間の幅）を大きくするのが目的なのだろうか。このへんの事情はよくはわからない。

タイヤが発揮する力と車体

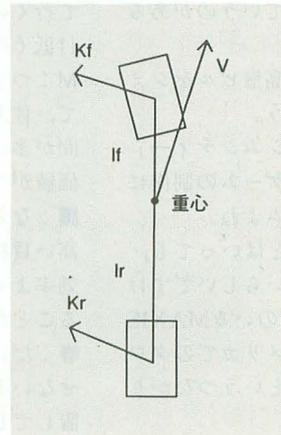
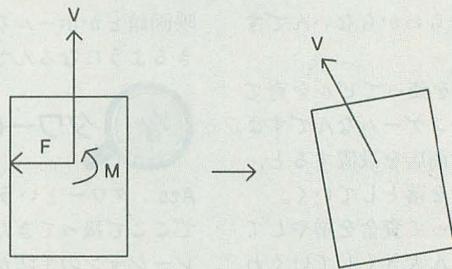
4輪が発揮する力と重心の運動の関係だが，実に複雑。律儀に考えると図4のように見たくもないような物理状態となる。ということで，当然考えやすいモデルとしたのが図5。まず，重心の位置を車体の中心線上と考える。こうすることで，前輪なら前輪，後輪ならば後輪の力の合計として2輪モデルとして扱うことができる。

タイヤの発揮する力によって，車体を回転させるモーメントが発生し，車体が旋回運動をすることになる。簡単にいえばこのとおりなのだが，そんな簡単ではない。モーメントで車体が回転するのは簡単なのだが，進行方向をどのように考えるかが困ったものである。もしこの2つを別の問題として扱うと，非常に困ったことが起こる。車体，速度ベクトルの回転が同時で，同じような角速度でない限りタイヤのグリップが抜けることになってしまう。図4のように，簡単にオーバーステア，そしてありえないようなアンダーステア（？）が生じてしまうのである。実際のアンダーステアとは，細かく分析すればこのような状況なのかもしれないが，シミュレートしたときを考えると，おそらく非現実的な動きになるはずである。

具体的な予想は今回はやめておこう。次回，アンダーとオーバーの力学的解析，そしていい加減な車体の運動を考えたいと思っている。

いよいよ力学解析を行わなければならない段階に入ってきた。このあたりがはっきり見えれば車を走らせることはある程度可能となる。でもやっぱりいろいろと細かい調整が必要だし，さまざまな影響を取り入れる段階でとてつもない労力が必要となるのは目に見えているが，まだベストの方法を探っている段階なので，よくわからない部分も多い。ということでまた来月。

図5　図4の簡略化モデル

図4　4輪が発揮する力と重心の運動の関係

ここで困るのは，合力とモーメントは独立して運動，回転を決定することである。

極端な場合，このようなありえないアンダーステアが起きてしまう。

それぞれの車輪にサイドフォースKf1,Kf2,Kr1,Kr2がかかる。

$F=Kf_1+Kf_2+Kr_1+Kr_2$

車体を回転させるモーメントは，

$M=Kf_1 \cdot lr+Kf_2l_4-Kr_1lr-Kr_2lr$

さらに，進行方向成分駆動，制動による合力でもモーメントは起こる。直線での急制度によるステアのとられ，スピンなどは，このモーメントによる。

こちらシステムX探偵事務所

FILE-XXIII

illustration：T.Takahashi

タワー型のシミュレーションモデル

Shibata Atsushi 柴田　淳

今回はMacintosh用のシミュレーションゲーム「タワー」について考察していきます。果たして，第3世代のシミュレーションは見つかるのでしょうか？　サンプルは迷路の最短経路を探すプログラムです。

柴田淳(以下Ats)：MacintoshやWindows用のゲームで，「タワー」っていうのがあるのを知ってますか？

マスター(以下M)：ああ，高層ビルをシミュレートするゲームでしょう。

琴張春香(以下春)：確か,「シムシティー」の作者のウィル・ライトがゲームの制作に加わっているとかいうゲームよね。

Ats：制作に加わっているとはいっても，ちょっと名前を貸したぐらいらしいですけどね。ただ，ウィル・ライトのいるMAXISというゲームメーカーがアメリカでのタワーの販売代理店をしてる，というつながりもあるみたいですけど。

琴張護(以下護)：タワーというゲームは，アメリカでは「シムタワー」という名前で売られていたような気がします。

M：ところで，タワーってどんなゲームなんですか？　名前は知っているけど，実際に遊んだことはないからわからないんですよ。

Ats：与えられた資金を使ってビルを育てていくシミュレーションゲームなんですよ。ビルに賃貸オフィスや商店を設置すると，人が集まってきてお金を落としていく。

M：なるほど。そうやって資金を増やしていって，ビルをどんどん大きくしていくわけですね。

Ats：大きくなっていくにつれて，ビルは高層化していきます。5階建てくらいなら階段を設置すればいいんですが，もっと階数が増えると，今度はエレベーターを設置しなければならなくなる。で，このエレベーターをいかに運用するかが，ゲームをうまく進めるカギになっているんです。

春：エレベーターをうまく運用する？

Ats：ええと，オフィスなどのビルの施設にはそれぞれ「価値」があるんです。この価値を決める大きな要因のひとつに，施設を使う人の「イライラ度」があります。このイライラ度は，ビルの住人が目的の階まで行くのにかかる時間が短ければ，それだけ低く抑えられるんです。

M：つまり，エレベーターの運用がまずくて，住人が目的の階まで行くときの待ち時間が多いと，イライラ度が高まって施設の価値が下がってしまうわけですね。

護：なるほど。価値が高い施設はそれだけ高い賃料を取れますから，エレベーターを効率よく運用できれば，より高い収益を得ることができる，というわけですね。

春：だけど，それなら住人をなるべく待たせないように，エレベーターをたくさん設置してしまえばいいんじゃないかしら。

Ats：ところが，そこはゲームらしく，ビルの規模に応じて，エレベーターの設置台数が制限されているんです。また，ビルの規模が大きくなっていけば，設置できる施設の種類も増えていきます。ゲームが進むと，映画館とかホールなんていう施設も設置できるようになるんですよ。

タワーのロジック

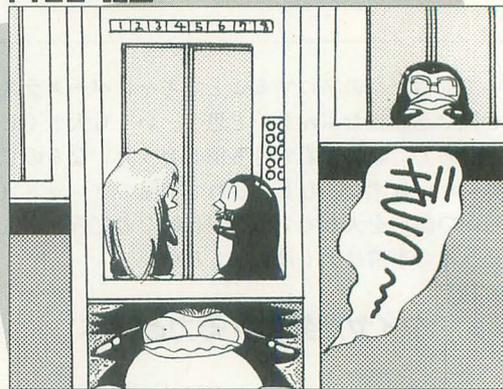
Ats：タワーというゲームですが，いままでここで扱ってきたものとは違ったシミュレーションの手法が使われているみたいなんです。

M：というと？

Ats：まず，オフィスや店舗などの施設の価値は，その施設にたどり着くまでの所要時間が長いか短いかで決まる，ということをいいましたよね。

春：施設を利用する住民のイライラ度が，施設の賃料に影響するんだったわよね。

護：すると，ゲームの内部では，ビルの施設ごとにイライラ度を数値として管理しているわけですね。

Ats：そうなんです。で，いままでここでやってきたような手法を使うとすると，このイライラ度は周りの状況から計算されるわけです。つまり，ビルのとある場所にあるオフィスまでの所要時間を，なんらかの擬似的な方法を使って裏マップで表現する，というような方法をとるんです。

M：そのような手法を使うと，常に最低限の部分の裏マップを書き換えるだけですむから，処理時間を抑えることができるんでしたよね。

Ats：ところが，タワーでは施設にたどり着くまでの時間を実際に人間を動かして計算しているんです。

M：確かに，ゲームの画面を見ると，たくさんの人間が動き回っていますね。

春：でも，こうやって住民を実際に動かすことによって，どうして施設に行くまでの時間を計れるの？

Ats：ええと，画面内を動いている住民ごとに，どの施設の利用者なのか，またどこに行こうとしているのかなどの個人情報が管理されているんです。この情報を元に，実際に住人を動かして，目的地に着くまでの所要時間を計って，目的地に着いたらイライラ度を管理する裏マップに所要時間を加算していくんです。

春：うーん，いかにも力業という感じね。

Ats：ただし処理速度のことを考えてか，住民を実際に移動させるといっても，杓子定規にやっているわけではなさそうなんです。たとえば，タワーではビル内の横の移動は考慮されていません。つまり，施設が横方向にいくら離れていようと，住民はまったくイライラしないんです。

護：しかし，それでもやはり実際に住民を移動させるというのは，かなりの処理速度を要求する処理であることには変わりないと思いますが。

M：そうまでしてこんな方法を使うというのは，それなりに理由があってのことなんでしょうね。

Ats：ひとつには，実際に人を動かすほうがはるかにリアルに見える，ということが大きな要因になっている気がします。
M：施設までの所要時間を表現するとしたら，たとえば施設ごとに便利さの裏マップを管理して，エレベーターに近いところにある施設は便利さを高く，遠いところは低くというふうにしてもいいわけですよね。
春：でも，デパートなんかのエレベーターを思い浮かべると，エレベーターの待ち時間ってもっと複雑な要素が絡んでいるんじゃない？
護：設置台数や収容人数，またエレベーターのカゴを運用するアルゴリズムなども関わってくるかもしれませんね。
M：すると，タワーというゲームはエレベーターという小道具をもち込もうと決めた時点で，実際に人を動かしてみるような手法をとらざるをえなくなった，といえるような気がしますね。

第3世代のシミュレーション

護：ところで，このタワーのような手法をシミュレーションのモデルとしてとらえると，どのようになるのでしょうか。
春：シミュレーションのモデルとしてとらえるって？
M：図1のように模式図化するとどうなるか，ということですよ。この図1は，確かいままでやってきたモデルの模式図でしたよね。裏マップを入力として，計算モデルを使って計算をして，その結果をさらに裏マップに反映させる，という意味じゃありませんでしたっけ。
Ats：そうですねぇ。まず，住民のイライラ度が計算モデルに入力されて施設の価値を決め，それがビルの収入になるんだから，図1で「入力」から右側は変わらないと思います。問題は入力の左側なんですが。
護：住民の移動にかかる時間が長いとイライラ度が増すのですから，入力に働くプラスの要因として，左側に住民の移動にかかる時間を書き込めばいいのではないでしょうか（図2）。
Ats：それはそうなんですけどね。さっきも同じ話が出ましたが，住民の移動時間を決める要因はかなり複雑ですよね。1次式で間に合うような単純な計算モデルとは次元が違いますよ。
M：それだけ複雑だからこそ，エレベーターの運用をシミュレートして，実際に住民を動かして移動に必要な時間を調べているわけですからね。
Ats：じゃあ，ここでいう住民の移動時間のように，詳細に調べてみないと決まらない要素は，計算モデルとは別扱いにしたほうがいいでしょうね。
M：こうして図にしてみると，2つのシミュレーションモデルの違いがはっきりしてきますね。
Ats：そうですね。便宜上，図1のようなモデルを「シムシティー型」，図2のようなモデルを「タワー型」としましょうか。
護：まず目につくのが，シムシティー型のモデルにはフィードバックがあり，タワー型にはない，ということでしょうか。
春：フィードバックがあるとないとでは，どんな違いが生まれるのかしら。
Ats：たとえば，シムシティーで町を育てていくとき，住宅地域に警察や学校を建設しないと，地域がスラム化するようなことが起こりますよね。つまり，安定しているように見える地域でも，フィードバックによって治安度の裏マップがだんだん悪化していき，放っておくと犯罪がはびこるようになるわけです。
M：すると，フィードバック機構をもっていないタワー型では，そのようなことが起こらないということですか？
Ats：厳密にいうと「起こりづらい」という感じかな。たとえば，タワーでビルの高層部に映画館のような大量の人間が出入りするような施設を作ってしまうと，エレベーターが混雑するようになって，結局映画館の近くの施設の価値は下がる，というようなことは起きるけど……。
護：しかし，それは単にエレベーターの利用頻度が高まるから，付近の施設へのアクセスが不便になっただけです。シムシティー型のスラム化のように「人口は増えたが悪いやつも増えた」というような，正負の要素が同居したタイプのフィードバックとは違うような気がします。
Ats：どちらかというと，僕もその考え方に賛成なんですよ。
M：あと，タワー型にあってシムシティー型にないのは，やっぱり実際に住民が動いている，というリアリティでしょうかね。
Ats：そうですね。シムシティー型のシミュレーションはかなり擬似的なものだから，たとえば住宅地域と工業地域が道路で結ばれていなくてもお互いに発展する，といったような，一見ウソっぽいところがどうしてもでてきてしまうでしょうね。
春：その点，タワー型では実際に住人をエレベーターに乗せて動かすのだから，エレベーターの通じていない階の施設に人が入るようなことはない，というわけね。
M：ところで，こんなふうに利点と欠点のある2つの手法なんだから，いっそお互いのいいところだけを組み合わせて新しい手法ができないものですかねぇ。
Ats：実は，タワーのマニュアルにウィル・ライトの文章が載っているんですけど，それを匂わすようなことが書いてあるんです。ウィル・ライトは最近「シムタウン」というゲームを出したようだから，そろそろタワー型とシムシティー型のハイブリッドシミュレーションを作りにかかっているのかもしれないですね。
春：そのハイブリッド型が第3世代の社会科学シミュレーションだ，というわけ？
Ats：うーん，既存のものを組み合わせただけのものが新しい世代か，というとちょっと疑問が残るかな。
M：じゃあ，第3世代が登場するのはまだ先だと。
Ats：いや，実はそれらしいアイデアは考えてあるんです。いままでのシミュレーションは，計算モデルがすべて固定されていて，そのためにダイナミックなゲーム展開

図1 シムシティー型のモデル

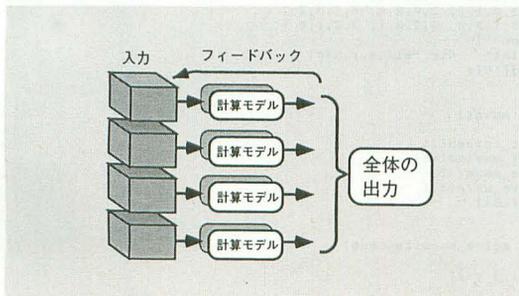
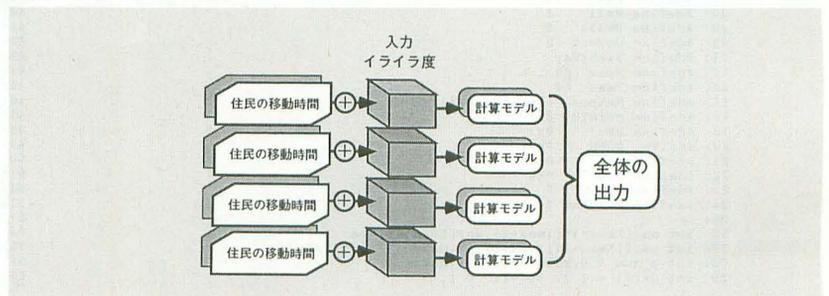



図2 タワー型のモデル

が望めませんでした。そこで，計算モデル自体を動的に割り当てるようにすればどうかなあ，と。
春：計算モデルを動的に割り当てる？
Ats：つまり，ゲーム進行によって計算モデル自体が変化していくようなモデル，ということになるかな。
M：それはまた，なんとも大胆な。
Ats：ただ，モデルのロジックは思いついたんですけど，うまい題材がなかなか見つからないんですよね。いずれにしろ，このアイデアが形になるにはもう少し時間が必要かな。
M：ロジックを思いついて，それを使うための適当な題材が見つからない，というのは，ずいぶんもったいない話ですね。
Ats：うーん，ロジックというか，シミュレーションのモデルを考えつくというのは，あくまでもゲームなどを作るときの必要条件の1つでしかないんですよ。アルゴリズムと適当な題材が結びついてはじめて，面白いゲームができあがるんだと思います。
春：なるほど。
Ats：また，モデルを考えつくだけでプログラミングに関しての仕事はすべておしまいか，というとそうでもないと思うんです。たとえば，タワーのエレベーター運行のシミュレーションのように，あまりにも専門化しすぎたものが関わってくると，ちょっとややこしくなる。
M：それはどういうことですか？
Ats：エレベーターの運行シミュレーションなんて，どちらかといえばアクションゲームで使われるような題材じゃないですか。
護：つまり，シミュレーションとはまったく別分野のものをプログラミングしなければならないわけです。ということは，通常より余分なノウハウが必要になる，ということですね。
Ats：そうなんですよ。仮に，タワーに登場する住人が，エレベーターの代わりに迷路状の通路を通って目的の施設に行く，というようなゲームを作るとしましょう。
M：そのゲームでは，入り口から施設までの最短経路の長さによって，住民のイライラ度が増減する，ということになるんですね。
Ats：ええ。このようなゲームを作るとすると，入り口から目的の施設までの，最短経路を求めるようなアルゴリズムを考え出さなければならないじゃないですか。
春：つまり，迷路を解くアルゴリズムが必要なわけね。なんだか難しそうだけど。
Ats：難しい，というより，基礎知識がないと解けない類の問題ですね。それほど大変そうではないから，ちょっと作ってみることにしましょうか。

サンプルについて

Ats：ええと，今回のサンプルでは，まず迷路を自動的に生成して，それから入り口から出口までの最短距離を求めます。
M：迷路を自動的に作るだけでも，けっこう大変そうだけど。
Ats：この手のアルゴリズムは腐るほどありますからね。興味のある人は参考文献を読んでみてください。
春：で，迷路はどうやって解くわけ？
護：普通の迷路では，常に迷路の壁に右手が触れるように道を通っていくと，いつかは必ず出口にたどり着ける，という法則を使うのではないでしょうか。
Ats：そうなんですよ。この法則を使うと，迷路を解く最短経路を簡単に見つけることができるんです。
M：もう少し具体的に教えてくださいよ。
Ats：まず，迷路を動く点を想定します。この点の右手が，常に壁につくように迷路内を移動させていきます。
春：でも，そうやって動いていると，いつかは袋小路にはいってしまうんじゃないかしら。
Ats：そうやって行き止まりに突き当たったら，今度は方向転換して点の軌跡を壁で埋めていくんです。こうやって袋小路をしらみつぶしに埋めて行って，最後に残ったのが出口への最短経路となるわけです。
春：なんだかさらりといってのけちゃったけど，やっぱり難しそうね。
Ats：でも，これくらいのアルゴリズムを軽くこなせるようでないと，タワー型のシミュレーションゲームは作れませんよ。
M：そうでしょうね。エレベーターの運行シミュレーションなどとなると，迷路を解くアルゴリズムよりもっと複雑になるでしょうから。
護：逆に，この種のシミュレーションとは一見関係ないようなノウハウをうまくゲームに取りこめれば，ゲームのリアリティは一段と増すでしょう。
Ats：シムシティー型のシミュレーションゲームでは，シミュレーションモデルの出来不出来がリアリティを左右するんでしょうが，タワー型では実際に人間を動かすわけですから，「人間らしさ」を演出することが，ゲームをリアルにするためには欠かせない要素だと思います。じゃあ，人間らしさとはなにかと考えてみると，やはり「頭のよさ」なんじゃないでしょうか？
M：つまり，タワー型の面白いシミュレーションゲームを作るためには，頭のいいアルゴリズムを作り出すノウハウが必要だ，ということですね。（つづく）

参考文献
奥村晴彦，「C言語による最新アルゴリズム辞典」，技術評論社

リスト1

```
 1:  /*  こちらシステムX探偵事務所*/
 2:  /*      1995年6号サンプルプログラム*/
 3:  /*      --  任意のキーで終了 --*/
 4: #include        "stdio.h"
 5: #include        "stdlib.h"
 6: #include        "basic0.h"
 7: #include        "BASIC.h"
 8: #include        "graph.h"
 9:
10: #define Way     0
11: #define Wall    1
12: #define PWall   2
13: #define Object  3
14: #define RightWay        4
15: #define Xmax  60
16: #define Ymax  60
17: #define MAXpos  (Xmax*Ymax/4)
18: #define cwidth  8
19: #define UP      0x1
20: #define DOWN    0x2
21: #define LEFT    0x4
22: #define RIGHT   0x8
23: #define STONE   5
24: #define DeadEnd 3
25:
26: int map[Xmax+1][Ymax+1],way[Xmax+1][Ymax+1];
27: int way2[Xmax+1][Ymax+1],xx[MAXpos],yy[MAXpos];
28: int p_num = 0,gole = 0,objs,cnt = 0;
29: int dx[4] = { 2, 0, -2,  0 };
30: int dy[4] = { 0, 2,  0, -2 };
31: int drx[4] = { 1, 0, -1,  0 };
32: int dry[4] = { 0, 1,  0, -1 };
33: int d_tab[24][4] = {
34:     0,1,2,3, 0,1,3,2, 0,2,1,3,
35:     0,2,3,1, 0,3,1,2, 0,3,2,1,
36:     1,0,2,3, 1,0,3,2, 1,2,0,3,
37:     1,2,3,0, 1,3,0,2, 1,3,2,0,
38:     2,0,1,3, 2,0,3,1, 2,1,0,3,
39:     2,1,3,0, 2,3,0,1, 2,3,1,0,
40:     3,0,1,2, 3,0,2,1, 3,1,0,2,
41:     3,1,2,0, 3,2,0,1, 3,2,1,0 };
42: struct  {
43:     int    dir,cell,x,y,atr;
44: } obj[16];
45:
46:
47: void main()
48: {
49:  init_screen();
50:  init_variables();
51:  make_maze();
52:  solve_maze(1);
53:  exit(0);
54: }
55:
56: void solve_maze(int num)
57: {
58: int i,j = 0;
```

```
 59:     while( gole < num ) {
 60:         for( i = 0; i < num; i++ ) {
 61:             if( obj[i].atr == -1 &&
 62:                 i*30 < cnt ) {
 63:                 obj[i].atr = 0;
 64:             }
 65:             move_object(i);
 66:         }
 67:         cnt++;
 68:         if( kbhit() ) {
 69:             break;
 70:         }
 71:     }
 72: }
 73:
 74: void move_object(int n)
 75: {
 76: int t,r,l,b,f;
 77:     if( obj[n].atr )
 78:     {
 79:             return;
 80:     }
 81:     f = obj[n].dir;
 82:     r = (obj[n].dir + 1)%4;
 83:     l = (obj[n].dir + 3)%4;
 84:     b = (obj[n].dir + 2)%4;
 85:     switch( way2[obj[n].x][obj[n].y] )
 86:     {
 87:         case 0 :
 88:             t = 1;
 89:             break;
 90:         case 1 :
 91:             if( !map[obj[n].x+drx[r]][obj[n].y+dry[r]] ) {
 92:                 t = 1;
 93:             } else {
 94:                 t = 0;
 95:             }
 96:             break;
 97:         case 2 :
 98:             if( !map[obj[n].x+drx[r]][obj[n].y+dry[r]] ) {
 99:                 t = 1;
100:                 break;
101:             }
102:             if( !map[obj[n].x+drx[f]][obj[n].y+dry[f]] ) {
103:                 t = 0;
104:                 break;
105:             }
106:             if( !map[obj[n].x+drx[l]][obj[n].y+dry[l]] ) {
107:                 t = 3;
108:                 break;
109:             }
110:             break;
111:         case 3 :
112:             if( !map[obj[n].x+drx[b]][obj[n].y+dry[b]] ) {
113:                 t = 2;
114:                 break;
115:             }
116:             if( !map[obj[n].x+drx[f]][obj[n].y+dry[f]] ) {
117:                 t = 0;
118:                 break;
119:             }
120:             if( !map[obj[n].x+drx[r]][obj[n].y+dry[r]] ) {
121:                 t = 1;
122:                 break;
123:             }
124:             if( !map[obj[n].x+drx[l]][obj[n].y+dry[l]] ) {
125:                 t = 3;
126:                 break;
127:             }
128:             break;
129:         default :
130:             break;
131:     }
132:     obj[n].dir = (obj[n].dir + t)%4;
133:     if( map[obj[n].x][obj[n].y] ) {
134:         draw_cell(obj[n].x,obj[n].y,
135:             map[obj[n].x][obj[n].y]);
136:     } else {
137:         draw_cell(obj[n].x,obj[n].y,RightWay);
138:     }
139:     if( way2[obj[n].x][obj[n].y] == 3 ) {
140:         map_to(obj[n].x,obj[n].y,PWall);
141:     }
142:     obj[n].x += drx[obj[n].dir];
143:     obj[n].y += dry[obj[n].dir];
144:     draw_cell(obj[n].x,obj[n].y,Object);
145:     if( obj[n].x > Xmax-2 ||
146:         obj[n].x > Ymax-3 ) {
147:         obj[n].atr = 1;
148:         gole++;
149:     }
150: }
151:
152: void make_maze()
153: {
154: int i,j,k,x,y,t,dd;
155:
156:     srand((unsigned)time(NULL));
157:     for (i = 4; i <= Xmax - 4; i += 2) {
158:         add_pos(i,2);
159:         add_pos(i,Ymax-2);
160:     }
161:     for (j = 4; j <= Ymax - 4; j += 2) {
162:         add_pos(2,j);
163:         add_pos(Xmax-2,j);
164:     }
165:     while( select_pos(&i,&j) ) {
166:         for ( ; ; ) {
167:             dd = 24*rnd();
168:             for (k = 3; k >= 0; k--) {
169:                 t = d_tab[dd][k];
170:                 x = i + dx[t];
171:                 y = j + dy[t];
172:                 if(map[x][y] == 0) {
173:                     break;
174:                 }
175:             }
176:             if(k < 0) {
177:                 break;
178:             }
179:             map_to((i+x)/2,(j+y)/2,1);
180:             i = x;
181:             j = y;
182:             map_to(i,j,1);
183:             add_pos(i,j);
184:         }
185:     }
186: }
187:
188: void add_pos(int i, int j)
189: {
190:     xx[p_num] = i;
191:     yy[p_num] = j;
192:     p_num++;
193: }
194:
195: int select_pos(int *i, int *j)
196: {
197: int r;
198:
199:     if ( !p_num ) {
200:         return( 0 );
201:     }
202:     p_num--;
203:     r = (int)(p_num * (rand() / (RAND_MAX + 1.0)));
204:     *i = xx[r];
205:     xx[r] = xx[p_num];
206:     *j = yy[r];
207:     yy[r] = yy[p_num];
208:     return( 1 );
209: }
210:
211: void map_to(x,y,a)
212: {
213:     if( x < 0 || x > Xmax ||
214:         y < 0 || y > Ymax ) {
215:         return;
216:     }
217:     map[x][y] = a;
218:     if( x > 1 && x < Xmax-1 &&
219:         y > 1 && y < Ymax-1 ) {
220:         draw_cell(x,y,a);
221:         change_way(x-1,y);
222:         change_way(x+1,y);
223:         change_way(x,y-1);
224:         change_way(x,y+1);
225:         change_way(x,y);
226:     }
227: }
228:
229: void change_way(int x,int y)
230: {
231: int i = 0;
232:
233:     if( map[x][y] ) {
234:         way2[x][y] = STONE;
235:         return;
236:     }
237:     if( map[x][y-1] ) {
238:         way[x][y] |= UP;
239:         i++;
240:     }
241:     if( map[x][y+1] ) {
242:         way[x][y] |= DOWN;
243:         i++;
244:     }
245:     if( map[x-1][y] ) {
246:         way[x][y] |= LEFT;
247:         i++;
248:     }
249:     if( map[x+1][y] ) {
250:         way[x][y] |= RIGHT;
251:         i++;
252:     }
253:     way2[x][y] = i;
254: }
255:
256: int draw_cell(int x,int y,int col)
257: {
258:     x -= 2;
259:     y -= 2;
260:     if( col < 16 ){
261:     fill(x*cwidth,y*cwidth,
262:         x*cwidth+cwidth-1,y*cwidth+cwidth-1,col) ;
263:     }
264: }
265:
266: int init_screen()
267: {
268: int i;
269:     screen( 2,0,1,1 ); console('NASI','NASI',0);
270:     palet(0,0 );
271:     palet(1,rgb(31,31,31) );
272:     palet(2,rgb(31,5,2) );
273:     palet(3,rgb(31,21,3) );
274:     palet(4,rgb(3,31,7) );
275: }
276:
277: int init_variables()
278: {
279: int i,j;
280:     for (i = 0; i <= Xmax; i++) {
281:         for (j = 0; j <= Ymax; j++) {
282:             map_to(i,j,1);
283:         }
284:     }
285:     for (i = 3; i <= Xmax - 3; i++) {
286:         for (j = 3; j <= Ymax - 3; j++) {
287:             map_to(i,j,0);
288:         }
289:     }
290:     map_to(2,3,0);
291:     map_to(Xmax-2,Ymax-3,0);
292:     for( i = 0; i <= 16; i++ ) {
293:         obj[i].x = 3;
294:         obj[i].y = 3;
295:         obj[i].dir = 0;
296:         obj[i].atr = -1;
297:     }
298: }
```

ローテク工作実験室　第9回

多チャンネルミキサを作ろう

Taki Yasushi 瀧　康史

X68000本体や各種MIDI音源，さらにはCD-ROMなど，私たちの扱う音楽ソースはどんどん増大しています。しかし，本格的なミキサでは大げさすぎるし……ということで今回は簡単なミキサを作ってみました。

出力装置はいっぱいある

X680x0シリーズはもともと音声出力を意識した作りになっているためか，背面にちゃんとAUXステレオ出力端子があります。一方，数年前からMIDIが盛んになってきて，MIDI楽器の音声もミキシングする必要が出てきています。最近ではSC-55をはじめとしてミキサ内蔵のMIDI楽器も増えてはきましたが，従来のCM-64，MT-32などを利用している人は別途ミキサを買わねばなりません。まあ，これだけの楽器構成ならば，CM-64＋(SC-55＋内蔵音源)という構成のため，4chミキサを買えば済みます。

2chでないのは，ミキサの多くが入力をモノラルで受け付けて出力をステレオ，つまり，右か左に振り分けるためです。多くは出力に「中ぐらい右」とか「わずかに左」とかの処置を行うために，可変抵抗(ボリューム)がつけられています。

ところで最近，X68000ユーザーのなかでもCD-ROMが流行しはじめました。一般のCD-ROMは，音楽CD再生のために音声出力があります。CD-ROMで音楽CDを再生するとデスクトップでCDの再生，停止などができますから，結構便利に使えます。それ以外にも，メガドライブのゆみみみっくすエミュレータをプレイする，Play Stationのぱずるだまエミュレータをプレイするなど，CDの音楽再生機能を利用するゲームなどの場合はミキシングは必須です。

これで，CD-ROM＋MIDIユーザの場合，6chミキサが必要になってきてしまいました。さらにAWESOME-XなどのPCM出力のできるボードを利用したりすると，8chミキサが必要になることになります。

本格的にコンピュータ音楽をするならば，大きなミキサを買えばよいでしょう。「専門的にやる人」用のミキサならば，値段，大きさ，機能，さまざまの商品があります。

しかし，CD-ROM＋MIDI楽器＋X680x0という組み合わせは，専門的に音楽をやる人以外でも割と持っている組み合わせです。安価でPCMボードが供給されたら，なおさらミキサが必要でしょう。

市販されている一般のミキサは，多チャンネルになればなるほど，たくさんの機能がついてきて，そして高価です。先ほど説明したような左右に振り分ける機能のほか，エフェクトへ送り返すような機能，入力インピーダンスをAUXレベルからMICレベルまで選択できる機能などいろいろとついているからです。「ただ単にミキシングしたいだけ！」といったケースの場合，なにも本格的なミキサを買う必要はありません。入力インピーダンスはAUXのみ，右左はなく，入出力はステレオ，エフェクトセンド，場合によっては，入力レベルなどもON/OFFにセレクトできる程度でよかったり，いらなかったりするかもしれません。

必要は発明の母。私のうちでは，X68000周りだけでも，CD-ROM，X680x0，CM-64，SC-55，実験中のPCMボードなど，ミキシングする必要のある機器があります。DSPボードが発売されたときのために，DAI to AUXのユニット，すなわちD/Aコンバータが作ってあり，これもミキシングせねばなりません(モノラル12chぐらい必要なのかな)。そこで，約束のSRAMボードをほっといて，多チャンネルミキサを作ってみることにしました。

これは完全に「ローテク」です。X680x0本体改造の必要性はありませんし，下手に作っても本体を壊すほどダメージを与える失敗はほとんどないでしょう。値段も，参考までに秋葉原価格ならば，あわせて2,000円前後です。

仕様の決定から完成まで，幅広いニーズにあわせて，原理を中心に説明しますから，ローテクしたい人。ぜひ挑戦してみてください。

仕様決定

まずは仕様を決定します。仕様決定に合わせて，ハードウェアの設計をするのです。きちんと考えておかないと，あとでハマるので注意しましょう。

●何チャンネルのステレオ入力があるのか？

まずミキサとしての第一条件ですね。何チャンネルステレオ入力があるのか？　今回は記事用として，ステレオ5ch(モノラル10ch)のミキサを作ってみることにしました。

配線や格納する箱などに拡張性を持たせておけば，多チャンネルになると思うので，いろいろ考えてみてください。

●各入力のボリュームを用意するか？

入力の抵抗を変えれば，レベルが変化します。選択は，固定抵抗，半固定抵抗，可変抵抗の3種類ぐらいあるでしょう。あんまり多機能だと配線がややこしくなりますし，大きな可変抵抗だと，まさしくボリュームになるので，パネルの加工も大変です。

各入力レベルは一度設定すれば，変更する必要はない……といい切れるなら固定抵抗がいいでしょう。そうすれば，回路は小さくまとまりますし，ミキサ本体が小さくなる分，ちょっとしたスペースにおけるという利点ができます。

一度設定したらほとんど変えないだろうけど，ひょっとしたら，気分でまた変えるかもしれない……と思うならば，半固定抵抗にするのがよいでしょう。こうした場合，パッケージを開けなくてはなりませんが，一応入力レベルを変えることができます。一度設定してみたものの，やっぱりもう少し内蔵音源を強くしたかった……なんて場合，すぐ変えられますしね。半固定抵抗は基板に実装するので，固定抵抗の場合よりも，基板そのものは大きくなります。

気まぐれなのでちょくちょく変える……という人なら，可変抵抗にするといいでしょう。この場合，ケースのパネルに可変抵抗をつけなくてはなりませんから，サイズも気にしなくてはなりません。基板そのものは線を引っ張るだけなので，小さくなるはずですが。

ボリュームを用意しなくても，ON/OFFの切り換えぐらいつけたい場合は，適当なスイッチをつけましょう。入力ラインを絶縁するだけなので，接続/絶縁の，2端子の簡単なスイッチでOKです。たとえば，X680x0などは膨大なノイズを出しますから，MIDI楽器のみの曲を鳴らしたい場合，本体側の入力を切ったほうがいいでしょう。

今回の記事中では，入力抵抗を固定，すべて同じにし，入力を切るスイッチはひとつだけ(X680x0本体用)にしてみることにしました。

部品選び

最初に決定されるのはオペアンプ(OP-Amp)です。これはNJM4580にて決定。オーディオ用といわれるOP-Ampで低雑音で低歪率，ヘッドフォン出力などの高出力もできるし，単一電源でも動作可能だからです。と，まあこういっておけば，かっこいいのですが，単に秋葉原の秋月電子で安売りしていたのが大きな原因です。

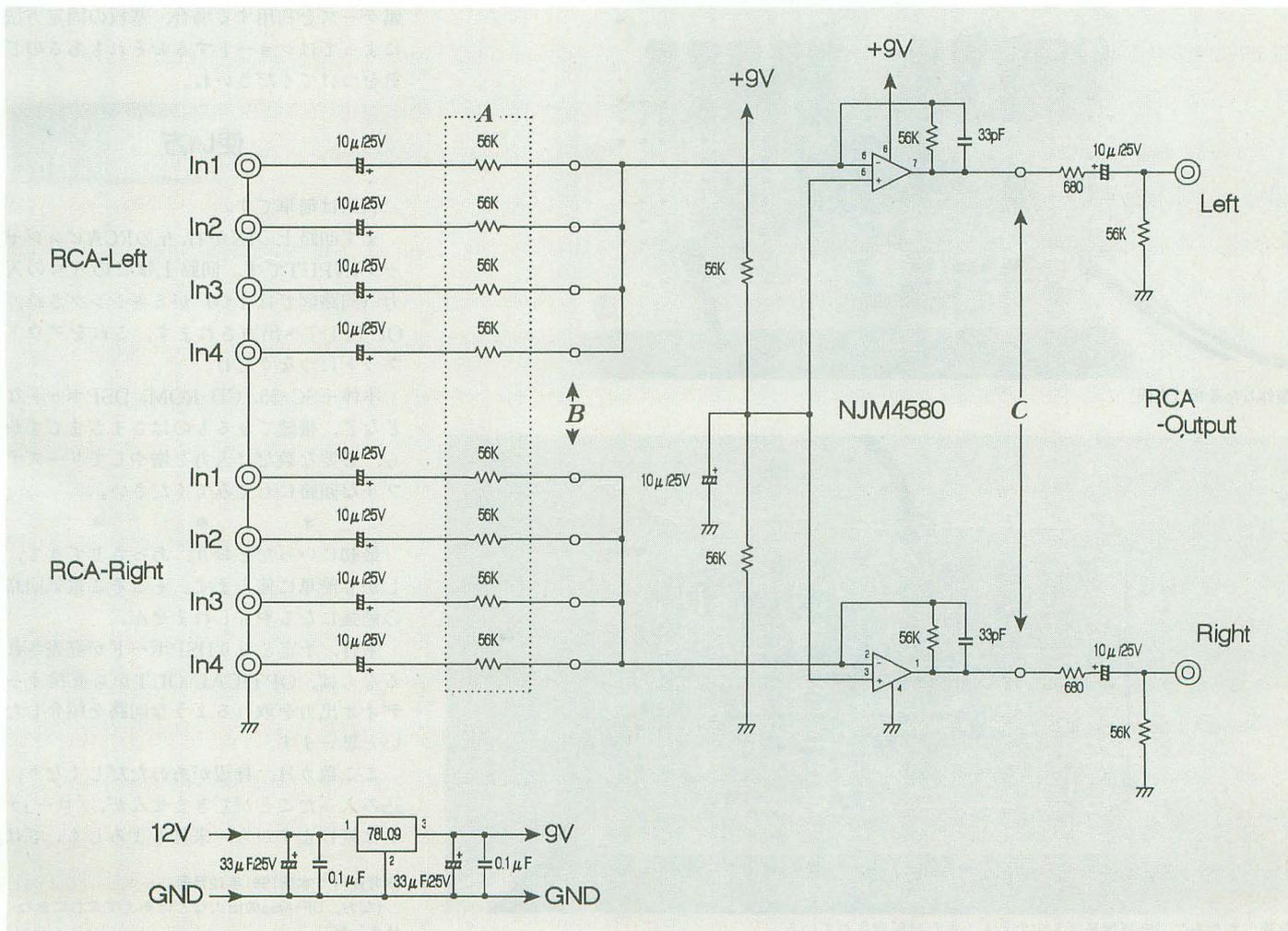
次に電源。電源はOP-Ampを駆動できればどういう奴でもよいのですが，電源が歪んでいれば出力も歪みます。したがって，ある程度は安定していなくてはなりません。スイッチング電源などを利用するのがリーズナブルですが，小さいスイッチング電源は高いという欠点があります。トランスを利用して電源を作ってもバチは当たりませんが，トランスを近くに置くと，配置をうまくやらないとノイズがバリバリ乗ります。

そういうわけで，アダプタです。アダプタを利用すると電源周りがゴツゴツするので好きじゃないんですが，「簡単ぽん」の利点は否定できないでしょう。もちろん，アダプタからくる電源は相当歪んでいますから，3端子レギュレータを利用して，適当な電源を作ります。そういうわけで，アダプタは12～15V程度のものを利用し，7809あたりを使って9Vを取り出すことにします。参考までに秋葉原の秋月電子でアダプタは300円にて購入，この端子を受けるコネクタは千石電商にて100円で購入。口径は2.1ϕです。また，7809は78L09(ローパワーだけど今回ぐらいの回路ならこれで十分)ですが，30円だったかな？　これもまた秋月電子です。

X680x0から＋12Vを引っ張ってきて，3端子レギュレータ78L05などを利用すれば，それなりに安定した電源を取ることもできますが，ミキサの電源はやっぱりつけっぱなしがよいのではないでしょうか？

原理説明

ミキサといったら反転増幅加算回路なのですが，電源を単一電源にしたため，負信号はそのままでは切れてしまいます。ということで，差動増幅回路にすることにしました。

図1　メイン回路図

基本回路は図1のとおりです。

図はステレオ4chですが，場合によって今回のような5ch，さらに6chなどを作りたい場合でも，入力を同様に増やすことによって処理できます(もっともあまり安定しないでしょうが)。

この図の囲みAは，入力インピーダンスの抵抗です。これを低くすると入力レベルを上げることができます。ただし，この抵抗が0になると，増幅率が理論上∞になるので，せいぜい2，3倍程度で20kΩから30kΩの抵抗をつけ，これと直列でなんらかの可変抵抗をつけてください。

もし入力信号のON/OFFをつけるならば，端子Bの部分で絶縁してください。

あと，人によってはもう少し出力がほしい場合があるかもしれません。そういうときには，もう1段なんらかのパワーアンプをつけるとよいでしょう。つける場所はCですね。Cに図2のような回路を含めると出力の増強ができると思います。

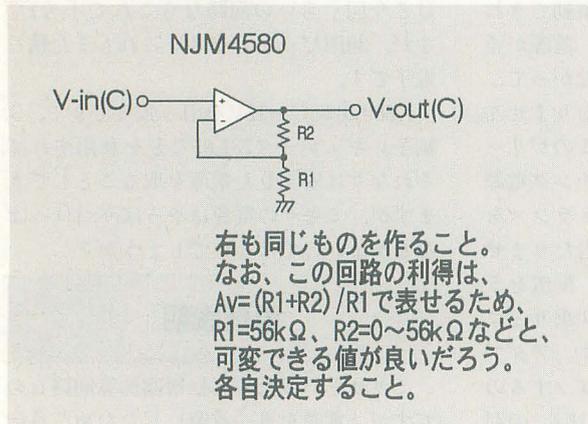
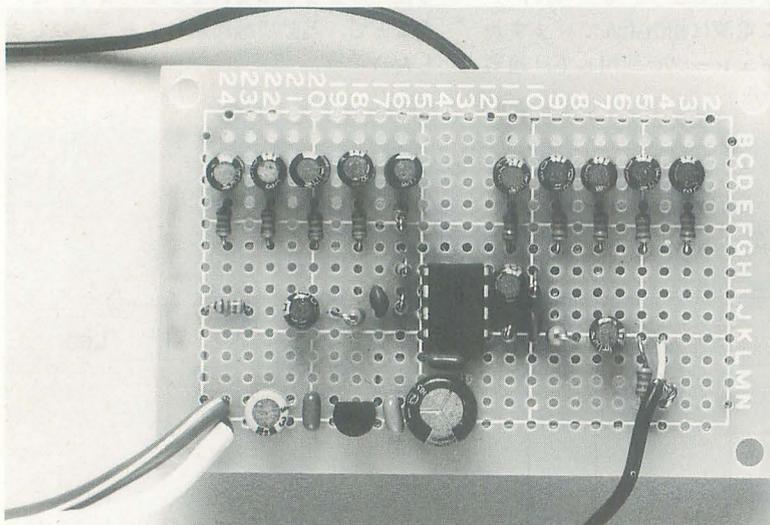
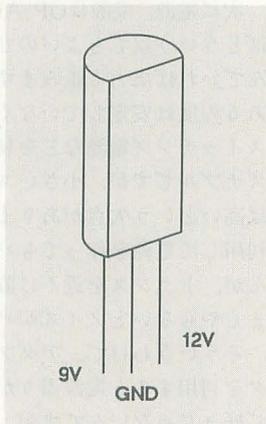
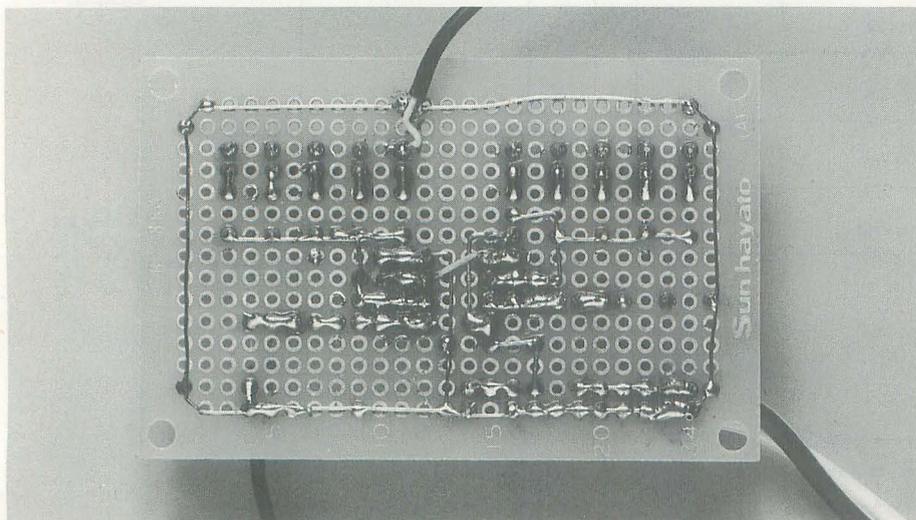
図2 パワーアンプの接続

図3 足に注意

図1下の9Vを生成する回路で間違いやすい点を。3端子レギュレータの足は正面(平らな面)から見て，順に1，2，3と数えます。詳しくは図3を参照のこと。

また，アダプタですが，＋－を接続前にきちんと確かめてください。アダプタのものによっては，中が－，外が＋になっていたりします。習慣で外は－などと思い込んでいると，ハマることになります。

電極を間違えると大電流が流れて危険ですからね。

なお，この回路を収めるケースはできる限り，金属ケースを利用してください。というのも，金属ケースのフレームをアースにすることができるからです。ただし，金属ケースを利用する場合，基板の固定方法によってはショートするおそれもあるので気をつけてくださいね。

製作した基板（表面）

裏面。ちなみに，音声信号は入出力とも1本だけ配線されている

使い方

利用は簡単です。

まず回路上のINが右，左のRCAピンジャックINPUTです。回路上はこの5つの入力（回路図では4つ）がミキシングされ，OUTPUTへ出力されます。これをアウトプットにつなぐだけ。

本体＋SC-55，CD-ROM，DSPボードなどなど，接続できるものはさまざまですから，必要な数だけ入力を増やしてリーズナブルな回路にしてみてください。

＊　＊　＊

最初にいったとおり，あっさりできて，しかも簡単に使えます。そこそこ電気回路の勉強になるやもしれません。

来月，予定どおりDSPボードが発売されるならば，OPTICAL OUTから直接オーディオ出力を取れるような回路を紹介したいと思います。

ここ数カ月，身辺があわただしくなり，込み入ったことができませんが，「ロー」テクは続けますから，来月もよろしく。では。

参考資料：本誌1991年12月号
（なお，OP-Ampの出力などはあえてこれにあわせました）。

〈対応機種一覧〉●MZ-80 K/C/700/1500 ●MZ-80 B/2000 ●MZ-2500/2861 ●X1 ●X1 turbo/Z ●PC-8001/8801/88 ●SMC-777/C ●PASOPIA/5 ●PASOPIA/7 ●FM-7/77/AV ●MSX/2/2+/turbo R ●PC-286/386/486/9801/98/9821 ●X68000/X68030
掲載されたプログラムの利用には各機種用のS-OS "SWORD" システムが必要です。

第156部 BLOCK DOWN

第157部 S-OSねちねち入門(3)

●BLOCK DOWN

今月は「S-OSねちねち入門(3)」とパズルゲーム「BLOCK DOWN」を掲載します。

「BLOCK DOWN」は，見た目が非常に簡単なパズルゲームです。実際に遊んでみても，なんの問題もなくルールを飲み込めるほどあまりにも単純なので，すぐに飽きるんじゃないかと考える人もいるかもしれません。確かに，なにげなくブロックを消しているだけではそれほど面白いゲームとはいえません。

しかし，高得点を狙い始めたとたんに，ゲームが熱くなります。得点の計算方法は，(消したブロックの個数$-2)^2$ですから，いっぺんに消すブロックが，多ければ多いほど獲得できるスコアが多くなります。したがって，プレイヤーは隣り合うブロックがなるべく多くなるように，脳味噌が加熱するまで考え込むことになるのです。うかつにブロックを消すと，まさにあちらが立てばこちらが立たず状態になってしまい，効率よくブロックを消すためにはかなりの思考力を必要とします。

さて，この単純ながら熱くなれる「BLOCK DOWN」。実は，フリーソフトウェアで出回っている「さめがめ」というパズルゲームが，元ネタとなっています。もともと，UNIX上で開発され，そのゲームの面白さに中毒者続出。現在では，MacintoshやPC-98，X68000にも移植され，さらにスーパーファミコンにまで移植が決定されるという人気ぶり（そのうちアーケード版として登場したりして）。ひと頃の「上海」のフィーバーを思い起こさせます。「BLOCK DOWN」は，画面いっぱいにブロックが置かれるところが他機種版と違いますが，シンプルな面白さは損なわれていません。リストも短めですので，ぜひハマってください。

●お絵描きパズル

「BLOCK DOWN」のような単純なパズルゲームは，S-OS "SWORD" の得意とするジャンルです。過去にも「COLUMNS」「PUSH BON!」などのパズルゲームが登場しています。そこで，次なるパズルゲームとして，S-OS "SWORD" 用「お絵描きパズル」なんてものはどんなものでしょう。これは，ヒントを頼りにマス目を埋めていくと，やがてドット絵になるというパズルです。ルールなどの詳しいことは，パズルゲーム雑誌に掲載されているので割愛させていただきますが，これもまた，じっくり考えながら結構ハマれるパズルなんですよね。

もちろん，エディタつきというのが条件です。自動解法プログラムを作るとなるとちょっとばかり面倒かもしれませんが，システム自体はものすごく簡単に作れるはずですし，基本的にドット絵はモノクロなので，データはかなり少なくてすみます。

ネックとなるのは，画面の縦幅が少ないということですか。ヒントとマス目を表示するとなると，25行ではそれほど大きなパズルを表示することができませんから。

それほど問題らしい問題もないので，できそうな気がするんですけど……どなたか挑戦しませんか？

●BREEZE

次に5月号で紹介した「Software DAC」の正式名称が「BREEZE」と決定したようです。作業も残すところ，ソースの統一化，ドキュメントの整理，サンプル，アプリケーションの制作，PSG出力テーブルをオリジナルのものに差し替え，開発スタッフ間の詳細事項のについての合意確認といったところまできている状況のようです。完成が非常に楽しみですね。

そして，5月号の記事でPSG再生プログラムの統一化が，奥野氏ひとりの手によって行われたかのように受け取れる記述がありました。実際は，保田周作，森喜一郎，筑紫高宏，奥野綾二(敬称略)，によって統一化が行われたことを改めて明記させていただきます。

1995■インデックス

全機種共通
S-OS"SWORD"要

BLOCK DOWN

Haruna Takayuki
春名 隆行

隣り合っているブロックを見つけて消していく。単純ながらついついのりめ込んでしまうパズルゲームです。できるだけ多くのブロックを消せるように，脳味噌をしぼるのだ。

私がこのゲームを知ったのは，某誌3月号の荻窪圭氏の記事によってです。ルールはシンプルなのに中毒者続出ということだったので，試しに作ってみるかという気になりました。

とりあえず，最初はX1のBASIC＋マシン語で作ったのですが，なかなか面白い！ それにこれぐらいならS-OS"SWORD"でも楽しめるんじゃないかと思い作成したのが，今回の作品です（オリジナルは「さめがめ」というタイトルでした）。

入力方法

まず，MACINTO-Cなどのマシン語入力ツールで，リスト1の3000_H～$33F3_H$まで打ち込んでください。チェックサムを確認したあと，いったんデバイスに，

#S BLD:3000:33F3:3000

のようにセーブしましょう。そして，

#J3000

でスタートします。

遊び方

さて，ゲームの遊び方ですが"2468"キーでカーソルの移動，"5"キーで2個以上隣り合っている（斜めはダメ）ブロックを一度に取ることができます。このとき，

(取ったブロック－2)^2

が得点となります。

また，取ったブロックのところにできる空間には，上のブロックが落ちてきます。縦一列すべて取ることができたときには，順次左側に詰められていきます。

このようにゲームを進めていくわけですが，ゲームの目的は2つあります。ひとつはすべてのブロックを取り除くことです。しかし，かなり難しく，運も必要でしょう。そして，もうひとつは，"ブロックを育てる"ことです。つまり，できるだけたくさんのブロックを一度に取れるように操作して，高得点を狙うというものです。どちらかといえば後者がメインの楽しみ方になると思います。

ちなみに，高得点を狙うコツとしては，ブロックを取ったときに，どのブロックが落ちてくるか的確に把握することです。慣れないうちはつらいかもしれませんが，先の先まで読めるようになることが高得点を獲得するための秘訣です。

そして，ゲームオーバー時には，スペースキーで再スタート，BREAKキーでモニタに戻ります。

プログラムについて

私はS-OS"SWORD"用のプログラムを組むのも，オールマシン語のプログラムを組むのも初めてだったのですが，シンプルなゲームだったためか，比較的短時間で完成しました。3400_H以降もワークとして使用していますので，注意してください。どこまで使用するかは，プレイヤーの腕と運しだいです。

なお，ソースはREDAで作成しましたが，文字列中にスペースのコード32を書き込んでいるところがあります。こうしないと，スペースを09_Hに変換してしまい正しくアセンブルできなかったのです（編注：これは，REDAのエディタ側の問題。タブ機能を実現するために，ソースコード中のスペースを強制的にタブコード09_Hに置き換えてしまうため起こるものです）。

そして，初期設定ではカーソル移動キーを"2468"，ブロックを取るキーを"5"としていますが，都合の悪い方は，

3116_H：上
3122_H：下
$311E_H$：左
$311A_H$：右
3126_H：ブロックを取る

以上のアドレスを書き換えてください。また，ブロックはわかりやすいものを選んだつもりですが，自分好みのブロックに変更したい方は，$33DF_H$からの5バイトを書き換えてください。

最後に

今回のプログラムは，できるだけ短くしようと心がけたのですが，いま考えると縦横のブロック数を変更できるようにしたほうがよかったかもしれませんね。

それでは，じっくり遊んでみてください。結構はまりますよ。

〈参考文献〉
Oh!X 1991年2月号　Z80's Bar

リスト1 BLOCK_DOWN.OBJ

```
3000 CD 22 30 CD 6F 30 CD 03 : 5B
3008 31 CD 54 31 38 F8 CD D2 : 52
3010 31 CD 46 32 CD 67 32 CD : A9
3018 BB 32 CD F9 32 CD 18 33 : FD
3020 18 E4 21 00 00 22 E7 33 : 59
3028 3E 0C CD F4 1F CD 30 20 : 47
3030 ED 5F 32 E9 33 21 09 00 : C4
3038 CD 1E 20 CD E2 1F 3C 3C : 51
3040 3C 20 20 20 42 4C 4F 43 : BC
3048 4B 20 44 4F 57 4E 20 20 : E3
3050 20 3E 3E 3E 00 21 1B 01 : 17
3058 CD 1E 20 CD E2 1F 48 49 : 6A
3060 5F 53 43 3A 30 30 30 30 : EF
3068 30 00 FD 21 F1 33 C9 21 : 5C
3070 00 00 22 E5 33 21 02 01 : 5E
3078 CD 1E 20 CD E2 1F 53 43 : 6F
----------------------------------
SUM: CA 68 1B 5A 8B 08 60 A6 6ECB

3080 3A 30 30 30 30 30 00 24 : 4E
3088 CD 1E 20 CD E2 1F 52 45 : 70
3090 53 54 3A 37 39 38 00 21 : AA
3098 1E 03 22 EB 33 DD 21 DF : 3E
30A0 33 21 01 03 22 F1 33 3E : DC
30A8 27 32 F3 33 0E 18 06 26 : D1
30B0 CD 1E 20 CD C1 30 CD F4 : 8A
30B8 1F 10 F8 24 79 BC 20 EE : 8E
30C0 C9 D9 2A E9 33 5D 54 CD : 66
30C8 F1 30 ED 5F AC AD 32 E9 : E1
30D0 33 CB 3C CB 1D CB 3C CB : F4
30D8 1D CB 3C CB 1D CB 3C CB : DE
30E0 1D 7D D9 E6 0E 0F D6 03 : 4F
30E8 38 D7 32 EF 30 DD 7E 00 : BB
30F0 C9 3E 10 44 4D 21 00 00 : C9
30F8 29 EB 29 EB 30 01 09 3D : 9F
----------------------------------
SUM: 0F 42 8B 28 BC 07 F4 3B BB1F

3100 20 F6 C9 11 03 31 D5 2A : 23
3108 F1 33 CD 1E 20 CD 21 20 : 3D
3110 FE 1B CA 70 33 FE 38 28 : E4
3118 1D FE 36 28 21 FE 34 28 : F4
3120 25 FE 32 28 27 FE 35 20 : F7
3128 E4 2A F1 33 CD 1B 20 FE : 38
3130 20 28 DA D1 4F C9 7C FE : 85
3138 03 C8 FD 35 01 C9 7D FE : 42
3140 26 C8 FD 34 00 C9 2D C8 : DD
3148 FD 35 00 C9 7C FE 17 C8 : 54
3150 FD 34 01 C9 06 00 11 00 : 12
3158 34 CD C7 31 1C ED 53 EF : 44
3160 33 CD 97 31 ED 5B ED 33 : 30
3168 2A EF 33 B7 ED 52 30 0D : 7F
3170 2A EF 33 5E 23 56 23 22 : 68
3178 EF 33 EB 18 E4 78 FE 02 : 81
----------------------------------
SUM: 22 36 3D 7D 3A D4 96 97 F7A5

3180 D8 CD C4 1F 32 E4 33 EB : BC
3188 56 2B 5E 2B EB CD 1E 20 : 00
3190 CD F1 1F 10 F2 B7 C9 2D : 8C
3198 CD AB 31 2C 2C CD AB 31 : AA
31A0 2D 25 CD AB 31 24 24 CD : 10
31A8 AB 31 C9 CD 1B 20 B9 C0 : 26
31B0 ED 5B ED 33 1A BC 1B 20 : 79
31B8 03 1A BD C8 1B 7A FE 34 : 69
31C0 30 F2 ED 5B ED 33 13 7D : 1A
31C8 12 13 7C 12 ED 53 ED 33 : 13
31D0 04 C9 3A E4 33 D6 02 C8 : BE
31D8 26 00 6F 54 5D CD F1 30 : 34
31E0 ED 5B E5 33 19 22 E5 33 : B3
31E8 01 99 33 C5 CD 15 32 D1 : 77
31F0 21 05 01 CD 1E 20 CD E5 : E4
31F8 1F 2A E7 33 ED 5B E5 33 : C3
----------------------------------
SUM: 2A 50 C4 96 17 8A 77 0E A0E7

3200 B7 ED 52 D0 ED 53 E7 33 : 20
3208 11 99 33 21 21 01 CD 1E : 0B
3210 20 CD E5 1F C9 11 10 27 : 02
3218 CD 3A 32 02 0C 11 E8 03 : 43
3220 CD 3A 32 02 0C 11 64 00 : BC
3228 CD 3A 32 02 0C 11 0A 00 : 62
3230 CD 3A 32 02 0C 3E 30 85 : 3A
3238 02 C9 AF ED 52 38 03 3C : 30
3240 18 F9 19 C6 30 C9 3A E4 : 07
3248 33 16 00 5F 2A EB 33 B7 : A7
3250 ED 52 22 EB 33 01 9B 33 : 4E
3258 C5 CD 25 32 D1 21 07 02 : E4
3260 CD 1E 20 CD E5 1F C9 01 : A6
3268 02 26 21 78 33 3E 01 77 : AA
3270 2C 10 FC ED 5B ED 33 1B : BB
3278 1A 47 21 77 33 85 6F 35 : 55
----------------------------------
SUM: 30 CD 9F F0 5D B3 C8 D4 D11F

3280 CC 8A 32 1B 7A FE 34 30 : 7F
3288 EE C9 D5 26 17 68 E5 11 : 27
3290 9F 33 CD 1B 20 FE 20 28 : 20
3298 04 12 1C 1C 1C 25 7C B9 : C4
32A0 20 F0 EB 16 20 7E A7 28 : 7E
32A8 06 72 2C 2C 2C 18 F6 E1 : EB
32B0 CD 1E 20 11 9F 33 CD E5 : A0
32B8 1F D1 C9 3A F3 33 47 0E : 6E
32C0 02 3D 6F 26 17 CD 1B 20 : F3
32C8 FE 20 CC D3 32 2D 7D A7 : 40
32D0 20 F3 C9 E5 11 78 33 E5 : 62
32D8 2C CD 1B 20 12 1C 7D B8 : 97
32E0 20 F6 AF 12 E1 CD 1E 20 : C3
32E8 11 78 33 CD E5 1F 25 7C : 2E
32F0 B9 20 E4 21 F3 33 35 E1 : 1A
32F8 C9 21 01 17 CD 1B 20 FE : 08
----------------------------------
SUM: 6E B5 D6 1A 9D 4D 46 FD 888E

3300 20 C0 21 0F 0B CD 1E 20 : 26
3308 CD E2 1F 41 4C 4C 20 20 : E7
3310 43 4C 45 41 52 00 18 3F : BE
3318 21 01 17 0E 02 3A F3 33 : A9
3320 47 CD 1B 20 FE 20 28 41 : D6
3328 57 25 CD 1B 20 BA C8 24 : 2A
3330 2C CD 1B 20 BA C8 7D B8 : EB
3338 20 E7 25 2E 01 7C B9 20 : B0
3340 E0 21 0E 0B CD 1E 20 CD : F2
3348 E2 1F 20 47 41 4D 45 20 : 5B
3350 20 4F 56 45 52 09 00 CD : 32
3358 D0 1F FE 1B 28 12 FE 20 : 60
3360 20 F5 21 06 30 E3 C3 6F : 81
3368 30 2C 7D B8 20 B3 18 CA : 46
3370 3E 0C CD F4 1F C3 FA 1F : 06
3378 00 00 00 00 00 00 00 00 : 00
----------------------------------
SUM: 7B 70 B1 8C 7B 50 A7 21 FC66

3380 00 00 00 00 00 00 00 00 : 00
3388 00 00 00 00 00 00 00 00 : 00
3390 00 00 00 00 00 00 00 00 : 00
3398 00 00 00 00 00 00 00 01 : 01
33A0 1D 1E 01 1D 1E 01 1D 1E : B3
33A8 01 1D 1E 01 1D 1E 01 1D : 96
33B0 1E 01 1D 1E 01 1D 1E 01 : 97
33B8 1D 1E 01 1D 1E 01 1D 1E : B3
33C0 01 1D 1E 01 1D 1E 01 1D : 96
33C8 1E 01 1D 1E 01 1D 1E 01 : 97
33D0 1D 1E 01 1D 1E 01 1D 1E : B3
33D8 01 1D 1E 01 00 00 00 23 : 60
33E0 2A 40 4F DB 00 00 00 00 : 94
33E8 00 00 00 00 00 00 00 00 : 00
33F0 00 00 00 00             : 00
----------------------------------
SUM: C0 F3 E6 71 96 79 95 BA 8950
```

リスト2 BLOCK_DOWN.S

```
0000                 1  ;****************************
0000                 2  ;  BLOCK DOWN
0000                 3  ;
0000                 4  ;   Programmed by  H A L
0000                 5  ;         '95/ 2/27
0000                 6  ;****************************
0000                 7     OFFSET  5000H
3000                 8     ORG      3000H
3000                 9  ;
1FFA P              10  #HOT       EQU      1FFAH
1FF4 P              11  #PRINT     EQU      1FF4H
1FF1 P              12  #PRINTS    EQU      1FF1H
1FE5 P              13  #MSX       EQU      1FE5H
1FE2 P              14  #MPRNT     EQU      1FE2H
1FD0 P              15  #GETKY     EQU      1FD0H
1FC4 P              16  #BELL      EQU      1FC4H
201B P              17  #SCRN      EQU      201BH
201E P              18  #LOC       EQU      201EH
2021 P              19  #FLGET     EQU      2021H
2030 P              20  #WIDCH     EQU      2030H
3000                21  ;
3000                22  ;ブロック座標格納用ワーク
3400 P              23  WORK_START EQU        $+400H
3000                24  ;
3000                25  ;  S T A R T
3000                26  ;
3000 CD 22 30       27     CALL      INIT1     ;画面作成&ワーク
3003 CD 6F 30       28     CALL      INIT2     ;画面作成&ワーク
3006                29  ;
3006                30  MAIN_LOOP:
3006 CD 03 31       31     CALL      CUR_MOVE     ;カーソル移動
3009 CD 54 31       32     CALL      CHECK        ;消えるか&消す
300C 38 F8          33     JR        C,MAIN_LOOP  ;消えないならループ
300E CD D2 31       34     CALL      SCORE_CALC   ;スコア計算&表示
3011 CD 46 32       35     CALL      REST_BLOCK   ;残りのブロック数表示
3014 CD 67 32       36     CALL      BLOCK_DOWN   ;下につめる
3017 CD BB 32       37     CALL      BLOCK_LEFT   ;左につめる
301A CD F9 32       38     CALL      CLEAR_CHECK  ;全部消えたか
301D CD 18 33       39     CALL      ALL_CHECK    ;まだ消せるか
3020 18 E4          40     JR        MAIN_LOOP
3022                41  ;
3022                42  ;
3022                43  INIT1:
3022 21 00 00       44     LD        HL,0
3025 22 E7 33       45     LD        (HI_SCORE),HL ;ハイスコアリセット
3028 3E 0C          46     LD        A,0CH
302A CD F4 1F       47     CALL      #PRINT       ;画面クリア
302D CD 30 20       48     CALL      #WIDCH       ;40文字
3030 ED 5F          49     LD        A,R
3032 32 E9 33       50     LD        (DATA),A     ;乱数初期値セット
3035 21 09 00       51     LD       HL,0009H
3038 CD 1E 20       52     CALL     #LOC
303B CD E2 1F       53     CALL     #MPRNT
303E 3C 3C 3C 20    54     DB        "<<<",32,"  BLOCK DOWN   >>>",0
3042 20 20 42 4C
3046 4F 43 4B 20
304A 44 4F 57 4E
304E 20 20 20 3E
3052 3E 3E 00
3055 21 1B 01       55     LD       HL,011BH
3058 CD 1E 20       56     CALL     #LOC
305B CD E2 1F       57     CALL     #MPRNT
305E 48 49 5F 53    58     DB       "HI_SC:00000",0
3062 43 3A 30 30
3066 30 30 30 00
306A FD 21 F1 33    59     LD       IY,CURSOR
306E C9             60     RET
306F                61  ;
306F                62  INIT2:
306F 21 00 00       63     LD       HL,0
3072 22 E5 33       64     LD       (SCORE),HL       ;スコア・リセット
3075 21 02 01       65     LD       HL,0102H
3078 CD 1E 20       66     CALL     #LOC
307B CD E2 1F       67     CALL     #MPRNT
307E 53 43 3A 30    68     DB       "SC:00000",0
3082 30 30 30 30
3086 00
3087 24             69     INC      H
3088 CD 1E 20       70     CALL     #LOC
308B CD E2 1F       71     CALL     #MPRNT
308E 52 45 53 54    72     DB       "REST:798",0
3092 3A 37 39 38
3096 00
3097 21 1E 03       73     LD       HL,21*38         ;残りブロック
309A 22 EB 33       74     LD       (REST),HL
309D DD 21 DF 33    75     LD       IX,CHR           ;キャラクタ・ワーク
30A1 21 01 03       76     LD       HL,0301H
30A4 22 F1 33       77     LD         (CURSOR),HL     ;カーソル座標(1,3)
30A7 3E 27          78     LD       A,39
30A9 32 F3 33       79     LD       (CH_RIGHT),A     ;最右端+1
30AC 0E 18          80     LD       C,24
30AE                81  SCR_MAKE1:
30AE 06 26          82     LD       B,38
30B0 CD 1E 20       83     CALL     #LOC
30B3                84  SCR_MAKE2:
30B3 CD C1 30       85     CALL       RND        ;A<=キャラクタ・コード
30B6 CD F4 1F       86     CALL     #PRINT
30B9 10 F8          87     DJNZ     SCR_MAKE2
30BB 24             88     INC      H
30BC 79             89     LD       A,C
30BD BC             90     CP       H
30BE 20 EE          91     JR       NZ,SCR_MAKE1
30C0 C9             92     RET
30C1                93  ;
30C1                94  RND:          ;Z80's Bar 第18回
30C1 D9             95     EXX        ;"乱数は世界を救うか"より
30C2 2A E9 33       96     LD         HL,(DATA)    ;平方採中法
30C5 5D             97     LD       E,L
30C6 54             98     LD       D,H
30C7 CD F1 30       99     CALL     KAKE
```

```
30CA ED 5F          100      LD      A,R
30CC AC             101      XOR     H
30CD AD             102      XOR     L
30CE 32 E9 33       103      LD      (DATA),A
30D1 CB 3C          104      SRL     H
30D3 CB 1D          105      RR      L
30D5 CB 3C          106      SRL     H
30D7 CB 1D          107      RR      L
30D9 CB 3C          108      SRL     H
30DB CB 1D          109      RR      L
30DD CB 3C          110      SRL     H
30DF CB 1D          111      RR      L
30E1 7D             112      LD      A,L
30E2 D9             113      EXX
30E3 E6 0E          114      AND       %00001110 ;下位３ビットでは、ばらつき
が良くない。
30E5 0F             115      RRCA                ;よって、ｂ１～ｂ３を使用。
30E6 D6 03          116      SUB     3
30E8 38 D7          117      JR      C,RND           ;Ａ <= ０～４
30EA 32 EF 30       118      LD      ($+5),A
30ED DD 7E 00       119      LD      A,(IX+0)
30F0 C9             120      RET
30F1                121  ;
30F1                122  KAKE:               ;ＨＬ <= ＨＬ＊ＤＥ
30F1 3E 10          123      LD      A,16
30F3 44             124      LD      B,H
30F4 4D             125      LD      C,L
30F5 21 00 00       126      LD      HL,0
30F8                127  ML1:
30F8 29             128      ADD     HL,HL
30F9 EB             129      EX      DE,HL
30FA 29             130      ADD     HL,HL
30FB EB             131      EX      DE,HL
30FC 30 01          132      JR      NC,ML2
30FE 09             133      ADD     HL,BC
30FF                134  ML2:
30FF 3D             135      DEC     A
3100 20 F6          136      JR      NZ,ML1
3102 C9             137      RET
3103                138  ;
3103                139  CUR_MOVE:
3103 11 03 31       140      LD      DE,$
3106 D5             141      PUSH    DE              ;ＲＥＴ用
3107 2A F1 33       142      LD      HL,(CURSOR)
310A CD 1E 20       143      CALL      #LOC
310D                144  CUR_M_LOOP:
310D CD 21 20       145      CALL      #FLGET
3110 FE 1B          146      CP        1BH     ;BREAK
3112 CA 70 33       147      JP        Z,END
3115 FE 38          148      CP        "8"     ;UP KEY CHECK
3117 28 1D          149      JR        Z,UP
3119 FE 36          150      CP        "6"     ;RIGHT KEY CHECK
311B 28 21          151      JR        Z,RIGHT
311D FE 34          152      CP        "4"     ;LEFT KEY CHECK
311F 28 25          153      JR        Z,LEFT
3121 FE 32          154      CP        "2"     ;DOWN KEY CHECK
3123 28 27          155      JR        Z,DOWN
3125 FE 35          156      CP        "5"     ;GET KEY CHECK
3127 20 E4          157      JR        NZ,CUR_M_LOOP
3129 2A F1 33       158      LD      HL,(CURSOR)
312C CD 1B 20       159      CALL    #SCRN
312F FE 20          160      CP      " "
3131 28 DA          161      JR      Z,CUR_M_LOOP
3133 D1             162      POP     DE
3134 4F             163      LD      C,A
3135 C9             164      RET
3136                165  UP:
3136 7C             166      LD      A,H
3137 FE 03          167      CP      3
3139 C8             168      RET       Z
313A FD 35 01       169      DEC       (IY+1)  ;Ｙ座標 －１
313D C9             170      RET
313E                171  RIGHT:
313E 7D             172      LD        A,L
313F FE 26          173      CP        38
3141 C8             174      RET       Z
3142 FD 34 00       175      INC       (IY+0)  ;Ｘ座標 ＋１
3145 C9             176      RET
3146                177  LEFT:
3146 2D             178      DEC       L
3147 C8             179      RET       Z
3148 FD 35 00       180      DEC       (IY+0)  ;Ｘ座標 －１
314B C9             181      RET
314C                182  DOWN:
314C 7C             183      LD        A,H
314D FE 17          184      CP        23
314F C8             185      RET       Z
3150 FD 34 01       186      INC       (IY+1)  ;Ｙ座標 ＋１
3153 C9             187      RET
3154                188  ;
3154                189  CHECK:
3154 06 00          190      LD      B,0
3156 11 00 34       191      LD      DE,WORK_START
3159 CD C7 31       192      CALL    ADR_STORE +5
315C 1C             193      INC     E
315D ED 53 EF 33    194      LD      (WORK_CH),DE
3161                195  CHECK_LOOP:
3161 CD 97 31       196      CALL    CHECK_MAIN
3164 ED 5B ED 33    197      LD      DE,(WORK_ED)    ;ワークの終りと、チェッ
ク済みのアドレスを
3168 2A EF 33       198      LD      HL,(WORK_CH)    ;比較する
316B B7             199      OR      A
316C ED 52          200      SBC     HL,DE
316E 30 0D          201      JR      NC,CLEAR_BL     ;全てチェック済みならば
、終り
3170 2A EF 33       202      LD      HL,(WORK_CH)
3173 5E             203      LD      E,(HL)
3174 23             204      INC     HL
3175 56             205      LD      D,(HL)
3176 23             206      INC     HL
3177 22 EF 33       207      LD      (WORK_CH),HL
317A EB             208      EX      DE,HL
317B 18 E4          209      JR      CHECK_LOOP
317D                210  ;
317D                211  CLEAR_BL:
317D 78             212      LD      A,B
317E FE 02          213      CP      2               ;２個以上？
3180 D8             214      RET     C
3181 CD C4 1F       215      CALL    #BELL
3184 32 E4 33       216      LD      (BLOCK),A
3187                217  CL_BL_LOOP:                 ;ワークから座標を取り出
し、スペースで消す
3187 EB             218      EX      DE,HL
3188 56             219      LD      D,(HL)
3189 2B             220      DEC     HL
318A 5E             221      LD      E,(HL)
318B 2B             222      DEC     HL
318C EB             223      EX      DE,HL
318D CD 1E 20       224      CALL    #LOC
3190 CD F1 1F       225      CALL    #PRINTS
3193 10 F2          226      DJNZ    CL_BL_LOOP
3195 B7             227      OR      A
3196 C9             228      RET
3197                229  ;
3197                230  CHECK_MAIN:                 ;ＨＬ＝中心座標
3197 2D             231      DEC     L               ;左右上下の順で調べる
3198 CD AB 31       232      CALL    CHECK_SUB
319B 2C             233      INC     L
319C 2C             234      INC     L
319D CD AB 31       235      CALL    CHECK_SUB
31A0 2D             236      DEC     L
31A1 25             237      DEC     H
31A2 CD AB 31       238      CALL    CHECK_SUB
31A5 24             239      INC     H
31A6 24             240      INC     H
31A7 CD AB 31       241      CALL    CHECK_SUB
31AA C9             242      RET
31AB                243  ;
31AB                244  CHECK_SUB:
31AB CD 1B 20       245      CALL    #SCRN
31AE B9             246      CP      C               ;同じ種類のブロック？
31AF C0             247      RET     NZ
31B0 ED 5B ED 33    248      LD      DE,(WORK_ED)
31B4                249  CH_SUB_LOOP:
31B4 1A             250      LD      A,(DE)          ;既にワークにある？
31B5 BC             251      CP      H
31B6 1B             252      DEC     DE
31B7 20 03          253      JR      NZ,CH_SUB2
31B9 1A             254      LD      A,(DE)
31BA BD             255      CP      L
31BB C8             256      RET     Z
31BC                257  CH_SUB2:
31BC 1B             258      DEC     DE
31BD 7A             259      LD      A,D
31BE FE 34          260      CP      WORK_START / 256
31C0 30 F2          261      JR      NC,CH_SUB_LOOP
31C2                262  ;
31C2                263  ADR_STORE:                  ;ワークに保存
31C2 ED 5B ED 33    264      LD      DE,(WORK_ED)
31C6 13             265      INC     DE
31C7 7D             266      LD      A,L
31C8 12             267      LD      (DE),A
31C9 13             268      INC     DE
31CA 7C             269      LD      A,H
31CB 12             270      LD      (DE),A
31CC ED 53 ED 33    271      LD      (WORK_ED),DE
31D0 04             272      INC     B
31D1 C9             273      RET
31D2                274  ;
31D2                275  SCORE_CALC:   ;SCORE<=(消えたブロック-2)^2
31D2 3A E4 33       276      LD      A,(BLOCK)
31D5 D6 02          277      SUB     2
31D7 C8             278      RET     Z
31D8 26 00          279      LD      H,0
31DA 6F             280      LD      L,A
31DB 54             281      LD      D,H
31DC 5D             282      LD      E,L
31DD CD F1 30       283      CALL    KAKE
31E0 ED 5B E5 33    284      LD      DE,(SCORE)
31E4 19             285      ADD     HL,DE
31E5 22 E5 33       286      LD      (SCORE),HL
31E8                287  SCORE_PRINT:                ;スコア表示
31E8 01 99 33       288      LD      BC,CH&PRNT +33
31EB C5             289      PUSH    BC
31EC CD 15 32       290      CALL    ASC_SET
31EF D1             291      POP     DE
31F0 21 05 01       292      LD      HL,0105H
31F3 CD 1E 20       293      CALL    #LOC
31F6 CD E5 1F       294      CALL    #MSX
31F9 2A E7 33       295      LD      HL,(HI_SCORE)
31FC ED 5B E5 33    296      LD      DE,(SCORE)
3200 B7             297      OR      A
3201 ED 52          298      SBC       HL,DE   ;ハイスコアより上？
3203 D0             299      RET     NC
3204 ED 53 E7 33    300      LD      (HI_SCORE),DE
3208 11 99 33       301      LD      DE,CH&PRNT +33
320B 21 21 01       302      LD      HL,0121H
320E CD 1E 20       303      CALL    #LOC
3211 CD E5 1F       304      CALL    #MSX
3214 C9             305      RET
3215                306  ;
3215                307  ASC_SET:      ;１０進ＡＳＣＩＩへ変換
3215 11 10 27       308      LD        DE,10000      ;万の位
3218 CD 3A 32       309      CALL      DIV
321B 02             310      LD      (BC),A
321C 0C             311      INC     C
321D 11 E8 03       312      LD      DE,1000         ;千の位
3220 CD 3A 32       313      CALL    DIV
3223 02             314      LD      (BC),A
3224 0C             315      INC     C
3225                316  ASC_SET2:
3225 11 64 00       317      LD      DE,100          ;百の位
3228 CD 3A 32       318      CALL    DIV
322B 02             319      LD      (BC),A
322C 0C             320      INC     C
322D 11 0A 00       321      LD      DE,10           ;十の位
3230 CD 3A 32       322      CALL    DIV
3233 02             323      LD      (BC),A
3234 0C             324      INC     C
3235 3E 30          325      LD      A,30H           ;一の位
3237 85             326      ADD     A,L
3238 02             327      LD      (BC),A
3239 C9             328      RET
323A                329  ;
323A                330  DIV:                ;A<=(HL/DE) のASCII
323A AF             331      XOR       A  ;HL<=HL-DE*A
323B                332  DIV_LOOP:
323B ED 52          333      SBC     HL,DE
323D 38 03          334      JR      C,CH_ASC
323F 3C             335      INC     A
3240 18 F9          336      JR      DIV_LOOP
3242                337  CH_ASC:
3242 19             338      ADD     HL,DE
3243 C6 30          339      ADD     A,30H
3245 C9             340      RET
3246                341  ;
3246                342  REST_BLOCK          ;残りブロック計算
3246 3A E4 33       343      LD      A,(BLOCK)
3249 16 00          344      LD      D,0
324B 5F             345      LD      E,A
```


▶テレビを見ていたら，「鶏の鳴き声はどのように聞こえるのか」ということで，各国の人に尋ねていた。これで気づいたのは，日本のように「コケコッコー」と聞こえる国はないということだった。　辻本　義成(32)三重県

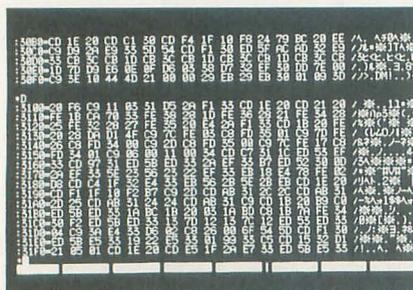
```
324C 2A EB 33     346      LD      HL,(REST)
324F B7           347      OR      A
3250 ED 52        348      SBC     HL,DE
3252 22 EB 33     349      LD      (REST),HL
3255 01 9B 33     350      LD      BC,CH&PRNT +35
3258 C5           351      PUSH    BC
3259 CD 25 32     352      CALL    ASC_SET2
325C D1           353      POP     DE
325D 21 07 02     354      LD      HL,0207H
3260 CD 1E 20     355      CALL    #LOC
3263 CD E5 1F     356      CALL    #MSX
3266 C9           357      RET
3267              358  ;
3267              359  BLOCK_DOWN   ;ブロックを下に詰める
3267 01 02 26     360      LD        BC,2602H
326A 21 78 33     361      LD      HL,CH&PRNT
326D 3E 01        362      LD      A,1
326F              363  BL_W_RES:
326F 77           364      LD      (HL),A
3270 2C           365      INC     L
3271 10 FC        366      DJNZ    BL_W_RES
3273              367  ;
3273 ED 5B ED 33  368      LD      DE,(WORK_ED)
3277              369  BL_DN_LOOP:  ;一度詰めたラインはとばす
3277 1B           370      DEC     DE
3278 1A           371      LD      A,(DE)
3279 47           372      LD      B,A
327A 21 77 33     373      LD      HL,CH&PRNT -1
327D 85           374      ADD     A,L
327E 6F           375      LD      L,A
327F 35           376      DEC     (HL)
3280 CC 8A 32     377      CALL    Z,DOWN_SUB
3283 1B           378      DEC     DE
3284 7A           379      LD      A,D
3285 FE 34        380      CP -    WORK_START /256
3287 30 EE        381      JR      NC,BL_DN_LOOP
3289 C9           382      RET
328A              383  ;
328A              384  DOWN_SUB:    ;ワークに縦一列分の表示データ作成
328A D5           385      PUSH    DE
328B 26 17        386      LD      H,23
328D 68           387      LD      L,B
328E E5           388      PUSH    HL
328F 11 9F 33     389      LD      DE,DOWN_WORK
3292              390  DN_SUB_LOOP:
3292 CD 1B 20     391      CALL    #SCRN
3295 FE 20        392      CP      " "
3297 28 04        393      JR      Z,DN_SKIP
3299 12           394      LD      (DE),A
329A 1C           395      INC     E
329B 1C           396      INC     E
329C 1C           397      INC     E
329D              398  DN_SKIP:
329D 25           399      DEC     H
329E 7C           400      LD      A,H
329F B9           401      CP      C
32A0 20 F0        402      JR      NZ,DN_SUB_LOOP
32A2 EB           403      EX      DE,HL
32A3 16 20        404      LD      D," "
32A5              405  DN_SPC:
32A5 7E           406      LD      A,(HL)
32A6 A7           407      AND     A
32A7 28 06        408      JR      Z,DN_PRNT
32A9 72           409      LD      (HL),D
32AA 2C           410      INC     L
32AB 2C           411      INC     L
32AC 2C           412      INC     L
32AD 18 F6        413      JR      DN_SPC
32AF              414  DN_PRNT:
32AF E1           415      POP     HL
32B0 CD 1E 20     416      CALL    #LOC
32B3 11 9F 33     417      LD      DE,DOWN_WORK
32B6 CD E5 1F     418      CALL    #MSX              ;表示
32B9 D1           419      POP     DE
32BA C9           420      RET
32BB              421  ;
32BB              422  BLOCK_LEFT:  ;縦一列全て無くなっていれば、左に詰め
る
32BB 3A F3 33     423      LD      A,(CH_RIGHT)
32BE 47           424      LD      B,A
32BF 0E 02        425      LD      C,2
32C1 3D           426      DEC     A
32C2 6F           427      LD      L,A
32C3 26 17        428      LD      H,23
32C5              429  BL_LT_LOOP:
32C5 CD 1B 20     430      CALL    #SCRN
32C8 FE 20        431      CP      " "
32CA CC D3 32     432      CALL    Z,BL_LT_SUB
32CD 2D           433      DEC     L
32CE 7D           434      LD      A,L
32CF A7           435      AND     A
32D0 20 F3        436      JR      NZ,BL_LT_LOOP
32D2 C9           437      RET
32D3              438  ;
32D3              439  BL_LT_SUB:
32D3 E5           440      PUSH    HL
32D4 11 78 33     441      LD      DE,CH&PRNT
32D7              442  LT_LOOP:
32D7 E5           443      PUSH    HL
32D8              444  LT_LOOP2:    ;ワークに横一列表示用データ作成
32D8 2C           445      INC     L
32D9 CD 1B 20     446      CALL    #SCRN
32DC 12           447      LD      (DE),A
32DD 1C           448      INC     E
32DE 7D           449      LD      A,L
32DF B8           450      CP      B
32E0 20 F6        451      JR      NZ,LT_LOOP2
32E2 AF           452      XOR     A
32E3 12           453      LD      (DE),A
32E4 E1           454      POP     HL
32E5 CD 1E 20     455      CALL    #LOC
32E8 11 78 33     456      LD      DE,CH&PRNT
32EB CD E5 1F     457      CALL    #MSX              ;表示
32EE 25           458      DEC     H
32EF 7C           459      LD      A,H
32F0 B9           460      CP      C
32F1 20 E4        461      JR      NZ,LT_LOOP
32F3 21 F3 33     462      LD      HL,CH_RIGHT       ;チェック用の右端を1減
らす
32F6 35           463      DEC     (HL)
32F7 E1           464      POP     HL
32F8 C9           465      RET
32F9              466  ;
32F9              467  CLEAR_CHECK: ;最左下にブロックがあるかでクリアチェ
ック
32F9 21 01 17     468      LD      HL,1701H
32FC CD 1B 20     469      CALL    #SCRN
32FF FE 20        470      CP      " "
3301 C0           471      RET     NZ
3302 21 0F 0B     472      LD      HL,0B0FH
3305 CD 1E 20     473      CALL    #LOC
3308 CD E2 1F     474      CALL    #MPRNT
330B 41 4C 4C 20  475      DB      "ALL",32," CLEAR",0
330F 20 43 4C 45
3313 41 52 00
3316 18 3F        476      JR      GAME_OVER
3318              477  ;
3318              478  ALL_CHECK:
3318 21 01 17     479      LD        HL,1701H  ;HL<=最左下の座標から調べる
331B 0E 02        480      LD      C,2
331D 3A F3 33     481      LD      A,(CH_RIGHT)
3320 47           482      LD      B,A
3321              483  ALL_CH_LOOP: ;上 or 右に、中心と同じブロックがある
か？
3321 CD 1B 20     484      CALL    #SCRN
3324 FE 20        485      CP      " "
3326 28 41        486      JR      Z,ALL_CH_SKIP
3328 57           487      LD      D,A
3329 25           488      DEC     H
332A CD 1B 20     489      CALL    #SCRN
332D BA           490      CP      D
332E C8           491      RET     Z
332F 24           492      INC     H
3330 2C           493      INC     L
3331 CD 1B 20     494      CALL    #SCRN
3334 BA           495      CP      D
3335 C8           496      RET     Z
3336 7D           497      LD      A,L
3337 B8           498      CP      B
3338 20 E7        499      JR      NZ,ALL_CH_LOOP
333A              500  ALL_CH_SUB:
333A 25           501      DEC     H
333B 2E 01        502      LD      L,1
333D 7C           503      LD      A,H
333F 20 E0        505      JR      NZ,ALL_CH_LOOP
3341              506  ;
3341 21 0E 0B     507      LD      HL,0B0EH          ;無い場合
3344 CD 1E 20     508      CALL    #LOC
3347 CD E2 1F     509      CALL    #MPRNT
334A 20 47 41 4D  510      DB      " GAME  OVER ",0
334E 45 20 20 4F
3352 56 45 52 09
3356 00
3357              511  GAME_OVER:
3357 CD D0 1F     512      CALL    #GETKY
335A FE 1B        513      CP      1BH               ;BREAK
335C 28 12        514      JR      Z,END
335E FE 20        515      CP      " "
3360 20 F5        516      JR      NZ,GAME_OVER
3362 21 06 30     517      LD      HL,MAIN_LOOP
3365 E3           518      EX      (SP),HL
3366 C3 6F 30     519      JP      INIT2             ;最初へ
3369              520  ALL_CH_SKIP:
3369 2C           521      INC     L
336A 7D           522      LD      A,L
336B B8           523      CP      B
336C 20 B3        524      JR      NZ,ALL_CH_LOOP
336E 18 CA        525      JR      ALL_CH_SUB
3370              526  ;
3370              527  END:
3370 3E 0C        528      LD      A,0CH
3372 CD F4 1F     529      CALL    #PRINT
3375 C3 FA 1F     530      JP      #HOT
3378              531  ;
3378              532  ;  W O R K   A R E A
3378              533  ;
3378              534  CH&PRNT:                       ;フラグ、表示データ用ワ
ーク
3378              535      DS      38
339E 00           536      DB      0
339F              537  DOWN_WORK:              ;縦一列表示データ用ワーク
339F 01 1D 1E 01  538      DB      1,29,30,1,29,30
33A3 1D 1E
33A5 01 1D 1E 01  539      DB      1,29,30,1,29,30
33A9 1D 1E
33AB 01 1D 1E 01  540      DB      1,29,30,1,29,30
33AF 1D 1E
33B1 01 1D 1E 01  541      DB      1,29,30,1,29,30
33B5 1D 1E
33B7 01 1D 1E 01  542      DB      1,29,30,1,29,30
33BB 1D 1E
33BD 01 1D 1E 01  543      DB      1,29,30,1,29,30
33C1 1D 1E
33C3 01 1D 1E 01  544      DB      1,29,30,1,29,30
33C7 1D 1E
33C9 01 1D 1E 01  545      DB      1,29,30,1,29,30
33CD 1D 1E
33CF 01 1D 1E 01  546      DB      1,29,30,1,29,30
33D3 1D 1E
33D5 01 1D 1E 01  547      DB        1,29,30,1,29,30
33D9 1D 1E
33DB 01 00 00 00  548      DB        1,0,0,0
33DF              549  CHR:                ;キャラクタ・データ
33DF 23 2A 40 4F  550      DB      "#*@O¤"
33E3 DB
33E4              551  BLOCK:              ;今消したブロック数
33E4 00           552      DB        0
33E5              553  SCORE:              ;スコア
33E5 00 00        554      DW        0
33E7              555  HI_SCORE:           ;ハイ・スコア
33E7 00 00        556      DW        0
33E9              557  DATA:               ;乱数用ワーク
33E9 00 00        558      DW        0
33EB              559  REST:               ;残りブロック数
33EB 00 00        560      DW        0
33ED              561  WORK_ED:            ;ブロック座標格納最後部アドレス
33ED 00 00        562      DW        0
33EF              563  WORK_CH:            ;次にチェックするブロック座標アド
レス
33EF 00 00        564      DW        0
33F1              565  CURSOR:             ;カーソル座標
33F1 00 00        566      DW        0
33F3              567  CH_RIGHT:           ;チェックする最右端
33F3 00           568      DB      0
33F4              569
```

▶セカンドマシン(愛人)として買ったDOS/V機が正妻の座を狙っています。がんばれ！X68000。
八木　明(29)神奈川県

全機種共通
S-OS"SWORD"要

S-OS ねちねち入門(3)

Chikushi Takahiro
筑紫 高宏

メモリマップから未公開のディスクI/Oの解析まで，今月もねちねちとS-OS"SWORD"を解説していきます。重要な情報が満載ですので，S-OSユーザーはしっかり目をとおしてください。

ディスクフォーマット追加事項

ちょっと余談になりますが，X1フォーマットのディスクのディレクトリ部分について，もう少し解説します。S-OS"SWORD"のみを使用しているときには参照する必要はありませんが，X1のディスクを操作する場合に参考にしてください。以下の情報は，X1turboZのマニュアルから抜粋＋αしたものです。

●ファイル属性（＋0）

ビット0　Bin形式ファイル（バイナリファイル）
ビット1　Bas形式ファイル（BASICの中間言語のテキスト）
ビット2　Asc形式ファイル（ASCIIファイル，そして64Kバイト以上のバイナリファイル）
ビット3　未使用（0を書きます）
ビット4　1で不可視になります。
ビット5　1でリードアフターライトON
ビット6　1でライトプロテクト
ビット7　1で下位ディレクトリ

属性の各ビットは，矛盾がない限り，複数指定可能です。ただし，通常，ビット0～2は複数指定されません。複数指定すると，別の（未定義）ファイル形式ということになります。

通常のバイナリファイルは，Bin形式ですが，64Kバイト以上のファイルだと，この形式で表現することができません。そのためAscファイル形式で，中身はバイナリファイルという使い方もされます(ただし256バイト単位)。当然，ファイルサイズや先頭アドレスなどの情報は，意味をもたなくなります。

なお，Asc形式ファイルのサイズですが，X1ではFATのみ意味をもちますが，S-OS"SWORD"ではBin形式ファイルと同じく，ディレクトリ上のサイズが有効となります。

階層化ディレクトリですが，BASIC上からMKDIRすると，属性が「1100000_B」になります。さらに，拡張子の部分が強制的に「DIR」となります。

ビット5のリードアフターライトに関しては，勉強不足でわかりません。

●先頭クラスタ番号（＋29～＋31）

＋29　上位
＋30　下位
＋31　中位

ディスクのクラスタは4Kバイトなので，1バイトあれば，1Mバイトまで表現できます。ちなみに，2HDの場合は2バイトを使用します。3バイト用意してあるのは，256Mバイト以上の大容量デバイスを考慮したときのためでしょう。

メモリマップ

標準的なS-OS"SWORD"内部のメモリマップを紹介します。

S-OSシステムの主な構造ですが，まずBIOSや各機種モニタが存在し，その呼び出しなどを共通化＋αしたものが，初代S-OS"MACE"です。多くのマシンのS-OSシステムでは，0000_H番地からBIOSやマシン語モニタが存在していて，そのすぐ後ろに付加する形で，基本的なサブルーチンの呼び出しの橋渡しをするようになっていました。ファイル操作は，カセットテープのみ対応しています。

S-OS"SWORD"へのバージョンアップは，主にディスクへの対応です。共通のディスク処理ルーチンと，原始的な機種依存のディスクI/Oルーチン（指定レコードを読み書きするだけ）が，完全に分離した形で制作されていたので，S-OS"MACE"への若干の手直しと，ディスクI/Oルーチンの制作だけで，各機種への移植を行うことができました。

S-OS"SWORD"を制作する場合，特に共通ディスク処理ルーチンを使用する場合は，プログラムアドレスなどに制限が出てきます。もちろん，手間はかかりますが，標準で用意されたものを使用しないことも可能です（要するにすべてを作り直せばいい)。その場合，アドレスの制約は若干減ります。ちなみに，S-OS"SWORD"システムが使用する0000_H～$2FFF_H$のエリアには，空きエリアがほとんどない状況で，S-OS"SWORD"からのバージョンアップが行われなかった大きな理由のひとつでしょう。

次に，メモリマップの内容を簡単に説明します（表1)。

S-OS"SWORD"のメイン部は，0000_H～$1CBF_H$にあります。BIOSを自作する場合でも，S-OS"SWORD"メイン部に使用できる領域は，7Kバイトほどです。

RUN&SUBMITは，バッチファイルの実行などの機能を追加するパッケージです。

次に位置するのが，S-OS"SWORD"のワークとジャンプテーブルです。通常のユーザーは，この部分を参照したり呼び出したりできます。

共通DOSモジュールは，ディスクの入出力を行うルーチンの固まりです。

ディスクI/Oルーチンは，DOSモジュールから呼び出されます。このルーチンを制作すれば，ディスクへの入出力が可能になります。一般的に，RAMディスクの処理ルーチン本体は，メモリの都合で2954$_H$～のエリアに配置されています。

ディスクI/O

ディスクI/Oは，S-OS“SWORD”でディスクを操作するための各機種依存部分です。ここは，共通DOSモジュールより呼び出されます。ディスクI/Oの仕様は，正式に公開されてはいないため，ソースリストを解析した結果を示します。

まず，ディスクI/Oのために用意されたエントリアドレスは，2B00$_H$～2B06$_H$の7バイトです。ディスクI/O本体は，2B07$_H$～2BFF$_H$に置かれます。ちなみに，ディスクI/Oルーチンは，リードとライトのルーチンを共通化して作成し，動作時に自己書き替えを行うことにより，サイズのコンパクト化を図っている場合が多いようです。

そして，RAMディスクのI/Oルーチンは，2B00$_H$～2B05$_H$のジャンプテーブルを乗っ取る形で動作します。このRAMディスクI/Oのためのルーチンは，だいたい空きエリアの2954$_H$～29FF$_H$に作成されています。

そして，ディスクI/Oのエントリアドレスは，次のようになっています。

DREAD（2B00$_H$）　ディスク読み込み
DWRITE（2B03$_H$）　ディスク書き込み
UNITNO（2B06$_H$）　ユニット番号（書き込みのみ）

S-OS“SWORD”では，フロッピーディスクは，4台までつなげられます。ドライブ名は“A～D”，ユニット番号は0～3に対応します。

さらに，ドライブ名“E”がRAMディスクとなっていて，ユニット番号が4となります。ディスクI/Oを呼び出す場合，対象となるドライブを，2B06$_H$にユニット番号で指定することになります。

なお，S-OS“SWORD”では，共通DOSモジュールのほうでも，ディスクI/Oでも，ベリファイを行っていません。

次に，ディスクI/Oの基本ルーチンの仕様について説明します。

●DREAD（2B00$_H$）

〈内容〉
ディスクからの読み込み
〈入力〉
HL←アドレス，DE←レコード番号，Acc←レコード数，[UNITNO]←ユニット番号
〈出力〉
Cy＝0（Acc←0）なら成功，Cy＝1（Acc←エラーコード）なら失敗
〈破壊〉
AF，AF´

●DWRITE（2B03$_H$）

〈内容〉
ディスクへの書き込み
〈入力〉
HL←アドレス，DE←レコード番号，Acc←レコード数，[UNITNO]←ユニット番号
〈出力〉
Cy＝0（Acc←0）なら成功，Cy＝1（Acc←エラーコード）なら失敗
〈破壊〉
AF，AF´

ちなみにこれらのルーチンは，S-OS“SWORD”のファンクションコールの#DRDSB（2000$_H$）と#DWTSB（2003$_H$）のメイン部分です。

次にディスクI/O関係の問題点を説明します。

まず，MZ-1500系統のディスクI/Oでは，割り込み禁止をしていないので，ディスクアクセス中に割り込みがかかった場合，エラーが発生する可能性があります。ただ，実際には，割り込みが使われないことが多いので，ほとんど問題となっていません。しかし，S-OS“SWORD”上でオリジナルのプログラム（S-OS“SWORD”に依存しないもの）を組もうとしているときは，注意してください。

そして，先ほど説明しましたが，ディスクI/OではAF´レジスタが破壊されます。さらに，それらを呼び出す共通DOSモジュールでも，AF´レジスタの保存を行っていません。したがって，ディスクをアクセスするルーチンを使用すると，AF´レジスタが破壊されてしまいます（仕様では保存されることになっている）。これは，バッチファイルを使用する場合にも当てはまります。つまり，ユーザーがディスクをアクセスするルーチンを使用する場合や，使用する可能性がある使い方をする場合，AF´レジスタは破壊されるとみなしてください。

また，これからディスクI/Oを作成する場合は，仕様書どおりAF´レジスタを保存するようにしましょう。

RAMディスクに関してですが，ETRK（20FF$_H$）というS-OS“SWORD”の内部ワークに，RAMディスクのクラスタ数を格納してあります。これにより，RAMディスクの有無，サイズを知ることが可能です。

おまけですが，X1用のS-OS“SWORD”では，ディスクのヘッドの移動速度を20msに設定してあります。しかし，内蔵ドライブは最高6msでもヘッドの移動が可能です。ということで，最高速に設定するための変更アドレスを書いておきます。ただし，ディスクのシーク音が気に入らない，ちょっとおかしいな？　と思ったらすぐに戻しましょう。

2C21：1E→1C
2C98：02→00

ちなみに，S-OS“SWORD”のディスクのデータバッファは，256バイトしか用意さ

表1　標準的なS-OS“SWORD”システムのメモリマップ

アドレス	内容
0000$_H$～	各機種BIOS，またはマシン語モニタ（各機種依存）
	S-OS“SWORD”メイン部（各機種依存）
～1CBF$_H$	バグ回避用空き（各機種依存）
1CC0$_H$～1F31$_H$	RUN&SUBMIT（各機種共通）
1F32$_H$～1F5A$_H$	RUN&SUBMIT空き
1F5B$_H$～1F81$_H$	S-OS“SWORD”ワーク
1F82$_H$～1F8D$_H$	S-OS“SWORD”ワーク拡張予約エリア
1F8E$_H$～2038$_H$	S-OS“SWORD”ジャンプテーブル
2039$_H$～204A$_H$	S-OS“SWORD”ジャンプテーブル拡張予約エリア
204B$_H$～20FE$_H$	バッチ変数
20FF$_H$	デバイス「E：」のクラスタ数
2100$_H$～28CC$_H$	共通DOSモジュール
28CD$_H$～28FF$_H$	共通DOSモジュール空き
2900$_H$～2953$_H$	S-OS“SWORD”内部ジャンプテーブル＆ワーク
2954$_H$～29FF$_H$	RAMディスクドライバ（機種依存）
2A00$_H$～2AFD$_H$	S-OS“SWORD”エラーメッセージ
2AFE$_H$～2AFF$_H$	エラーメッセージ空き
2B00$_H$～2CFF$_H$	ディスクI/Oルーチン（各機種依存）
2D00$_H$～2DFF$_H$	バッチファイル用ディスクバッファ
2E00$_H$～2EFF$_H$	ディスク入出力バッファ（FATバッファ）
2F00$_H$～2FFF$_H$	ディスク入出力バッファ（データバッファ）

れていません。このためセクタの大きさが，512バイトや1024バイトのディスクを読もうとすると，ユーザーエリアに重なるため，暴走する可能性があるので注意してください。

RUN&SUBMIT

RUN&SUBMITは，バッチファイルを実行可能にする共通パッケージです。これは，$1CC0_H$～$1F5A_H$に割り当てられています。#GETL($1FD3_H$)と#FLGET(2021_H)を乗っ取ります。ついでに，DOSモジュールのコマンドも拡張します。拡張されるコマンドは以下の2つです。

●；

"；"のある1行のコメントを表示します。MS-DOSのREMコマンドに相当するものです。

●P

バッチファイル中にキー入力待ちを行います。なにかキーを押せば，バッチファイルの実行を再開します。ただし，$1B_H$のキー(機種により，ESCやSHIFT＋BREAKなど）を押すと，バッチファイルの実行を中断します。

●RUNコマンド

RUN&SUBMITを組み込まない場合，ファイルの実行は，ロードコマンドでロードして，ジャンプコマンドで実行アドレスへジャンプします。S-OS"SWORD"では，コマンドラインで入力されたコマンドの前にあるスペースを取り除いて，コマンドの解析を行いません。つまり，コマンドの前にスペースを挿入してしまうと，コマンドを受けつけなかったのです。RUN&SUBMITでは，この性質を逆手に取って，スペースのあとにファイル名を続けると，そのファイルを実行するようにしています。

そしてアプリケーション側に，パラメータを渡すことも可能です。ファイル名のあとに"："をつけると，その次のアドレスをDEレジスタにセットして，アプリケーション側へ処理を移します。なにもパラメータがない場合には，DEレジスタに00_Hが入っています。

RUNコマンド（ファイルの自動実行）によりプログラムを実行し，終了した場合，再びRUN&SUBMIT側に制御が戻り，AFレジスタの内容を参照します。Cy＝1ならエラーで，Accの内容がエラー番号となります。もし，Cy＝1だったら，対応するエラーメッセージを表示します。よって，ユーザーがS-OS"SWORD"用プログラムを作成する場合，特に意図がなければ，Cy＝0，Acc＝0にしてから，プログラムを終了するようにしてください（プログラムの最後に"SUB A"の1行を加えればOKです)。

●バッチファイル

RUNコマンドで指定されたファイルがASCIIファイルなら，バッチファイル（S-OS"SWORD"では，キー入力ルーチンを乗っ取り，ファイルにある文字列をキー入力ルーチンに渡すようにしています）として実行されます。このバッチファイル読み込み用に，$2D00_H$～$2DFF_H$が使用されています。

バッチファイル中で"¥"（S-OS"SWORD"文字で"＼")は，バッチ変数のために使用されます。"¥"という文字を使いたい場合は，"¥"を2重の"¥¥"にすると，1文字の"¥"とみなされます。

●バッチ変数

MS-DOSなどにある，バッチ変数が使用可能になります。ただし，MS-DOSでは"%n"ですが，S-OS"SWORD"では，先ほどもいったように"¥n"です（S-OS"SWORD"では，キャラクタが一部違うので，実際の表示は"＼n"などになります)。"n"の部分は数字で1～9が指定可能です。1つの変数に20バイトが割り当てられ，19文字まで有効です。S-OS"SWORD"では，バッチ変数0番"¥0"を指定した場合ビープ音が鳴り，カーソル位置はそのままでユーザー側に制御が戻るようになっています。

このバッチ変数ですが，コマンドラインで指定されたスペースで区切られたパラメータを，順番に1～9と番号をつけてバッファに格納しておき，バッチファイル中にバッチ変数がくると，バッチ番号が格納された文字列に置き換えられます。

ちなみに，S-OS"SWORD"ではファイル名中にスペースを許している関係で，ファイル名（実行ファイル）とパラメータを"："で区切ってください。

それでは，RUN&SUBMIT関係の問題点などを説明します。

RUNコマンドを実行しようとした場合，ファイル先頭が3000_Hで，かつ，ファイルの大きさが7Kバイト未満なら，3000_H～$4BFF_H$のメモリの内容が，いったんRAMディスクの一部のエリアに待避されて，ファイルの実行後，元に戻されます。この待避用に使われるのが，RAMディスクのレコード0～13，18～31です。このうち，レコード18～31は，ディレクトリエリアなので，16個より多くのファイルが存在していた場合，重複しているファイルが破壊されることになります。ただし，RAMディスクの大きさが68Kバイト以内なら，この問題は発生しません。

もし，320KバイトのEMMなどを使用していて，16個以上にファイルを保存するつもりなら，以下の改造で，待避機能を潰し，問題を回避することが可能です。

1CFC:CD 15 1D→00 00 00

そして，画面2行にまたがるバッチファイルは，誤動作します。これは，論理桁位置の代わりにX座標を参照しているためです。

バッチファイル実行中は，S-OS"SWORD"ファンクションの#GETLや#FLGETで，AF'レジスタが破壊されることがあります（ディスクI/Oの関係上)。

また，起動時に実行されるAUTOEXEC.BATは，ドライブAに固定されています。立ち上がったドライブをデフォルトドライブにできるS-OS"SWORD"でも，強制的にドライブAに読みにいきます。デフォルトドライブよりAUTOEXEC.BATを実行したい場合は，以下のアドレスを変更してください。

1F1A:'A:AUTOEXEC.BAT',00_H→'AUTOEXEC.BAT',00_H,00_H,00_H

エラーコード

S-OS"SWORD"のエラー番号と，メッセージの対応表を掲載します(表2)。メッセージは，$2A00_H$～$2AFF_H$の共通DOSモジュールのソースに入っています。

エラー0は，エラーなしという意味で，エラー15以上だとそのエラー番号，たとえば，エラー15の場合「Error $0F」と表示されます。

表2 エラーコード

番号	メッセージ
1	Device I/O Error
2	Device Offline
3	Bad File Descripter
4	Write Protected
5	Bad Record
6	Bad File Mode
7	Bad Allocation Table
8	File not Found
9	Device Full
10	File Already Exists
11	Reserved Feature
12	File not Open
13	Syntax Error
14	Bad Data
15～	Error $??

コマンドライン

S-OS“SWORD”を立ち上げると，プロンプトの「#」記号が出て，コマンド入力待ちになります。この処理を行っているのが，共通DOSモジュールです。このとき使える内部コマンドを紹介します（dはデバイス名）。内部コマンドは，ほとんどが1文字（一部2文字）です。そして，コマンドのあとには，1文字以上のスペースを空けるのが望ましいとされています。

●L d:filename.ext
L d:filename.ext:nnnn

ファイルをロードします。

16進4桁でアドレスを指定すると，そのアドレスから読み込みます。

●S d:filename.ext:xxxx:yyyy
S d:filename.ext:xxxx:yyyy:zzzz

メモリの内容をセーブします。

アドレスは16進4桁で指定します。「xxxx」は先頭アドレス，「yyyy」は最終アドレス，「zzzz」は実行アドレスです。実行アドレスを省略すると，先頭アドレスが実行アドレスになります。

●K d:filename.ext

指定したファイルを消去します。

●N d:file1.ext:file2.ext

「file1.ext」を「file2.ext」にリネームします。

●ST d:filename.ext:P
ST d:filename.ext:R

ライトプロテクトの設定，解除を行います。パラメータに“P”を指定すると，ライトプロテクトがかかり，“R”を指定すると，ライトプロテクトを解除します。

●DV d:

デフォルトデバイスを変更します。

S-OS“SWORD”で，通常使用されるデバイスは，表3のとおりです。

●D

ディレクトリを表示します。

●!

ブートします。

●Jnnnn

16進4桁「nnnn」で指定したアドレスをコールします。

共通DOSモジュールの該当する処理ルーチンでは，DEレジスタを破壊してしまうので，パラメータを渡すことができません（このことは，深谷崇氏に教えていただきました）。パラメータを受け取ろうとすると，DEレジスタのアドレスが不定のため誤動作する可能性があります。以下の改造を行うと，“J”コマンドで，アプリケーションにパラメータを渡すことができるようになります。

```
217B:EB 21 00 21 E3 EB E9
        →E3 21 00 21 E3 E9 00
```

●M

各機種のモニタをコールします。

一般的なモニタでは，そのモニタの「R」コマンドなどで，呼び出したシステムに戻ってくることが可能ですが，一部の機種では戻る機能がありません。その場合は，S-OS“SWORD”の#COLD（$1FFD_H$）か，#HOT（$1FFA_H$）へジャンプしてください。

●W

40桁モードと80桁モードを切り替えます。「W」コマンドが実行されたとき，80桁モードなら40桁に，40桁モード以外なら80桁に変更されます。

表3 S-OS“SWORD”で使用されるデバイス

A	フロッピーディスク
B	フロッピーディスク
C	フロッピーディスク
D	フロッピーディスク
E	RAMディスク
Q	QD（処理的にはカセットテープ）
S	各機種依存フォーマットのカセットテープ
T	S-OS“SWORD” 2400bpsのカセットテープ

▶ 全機種共通システムインデックス ◀

＊以下のアプリケーションは，基本システムであるS-OS“MACE”またはS-OS“SWORD”がないと動作しませんのでご注意ください。

1987

■87年1月号
第35部 マシン語入力ツールMACINTO-C
連載 FuzzyBASIC料理法<4>
■87年2月号
第36部 アドベンチャーゲームMARMALADE
第37部 テキアペ作成ツールCONTEX
■87年3月号
第38部 魔法使いはアニメがお好き
第39部 アニメーションツールMAGE
付録 "SWORD"再掲載とMAGICの標準化
■87年4月号
第40部 INVADER GAME
第41部 TANGERINE
■87年5月号
第42部 S-OS "SWORD" 変身セット
第43部 MZ-700用 "SWORD" をQD対応に
■87年6月号
インタラプト コンパイラ物語
第44部 FuzzyBASICコンパイラ
第45部 エディタアセンブラZEDA-3
■87年7月号
第46部 STORY MASTER
■87年8月号
第47部 パズルゲーム碁石拾い
第48部 漢字出力パッケージJACKWRITE
特別付録 FM-7/77版S-OS "SWORD"
■87年9月号
第49部 リロケータブル逆アセンブラInside-R
特別付録 PC-8001/8801版S-OS "SWORD"
■87年10月号
第50部 tiny CORE WARS
第51部 FuzzyBASICコンパイラの拡張
第52部 X1turbo版S-OS "SWORD"
■87年11月号
序論 神話のなかのマイクロコンピュータ
付録 S-OSの仲間たち
第53部 もうひとつのFuzzyBASIC入門
第54部 ファイルアロケータ＆ローダ
インタラプト S-OSこちら集中治療室
第55部 BACK GAMMON
■87年12月号
第56部 タートルグラフィックパッケージTURTLE
第57部 X1turbo版 "SWORD" アフターケア
ラインプリントルーチン
特別付録 PASOPIA7版S-OS "SWORD"

1988

■88年1月号
第58部 FuzzyBASICコンパイラ・奥村版
付録 石上版コンパイラ拡張部の修正
■88年2月号
第59部 シューティングゲームELFES
■88年3月号
第60部 構造型コンパイラ言語SLANG
■88年4月号
第61部 デバッギングツールTRADE
第62部 シミュレーションウォーゲームWALRUS
■88年5月号
第63部 シューティングゲームELFES II
第64部 地底最大の作戦
■88年6月号
第65部 構造化言語SLANG入門(1)
第66部 Lisp-85用NAMPAシミュレーション
■88年7月号
第67部 マルチウィンドウドライバMW-1
連載 構造化言語SLANG入門(2)
■88年8月号
第68部 マルチウィンドウエディタWINER
■88年9月号
第69部 超小型エディタTED-750
第70部 アフターケアWINERの拡張
■88年10月号
第71部 SLANG用ファイル入出力ライブラリ
第72部 シューティングゲームMANKAI
■88年11月号
第73部 シューティングゲームELFES IV
■88年12月号
第74部 ソースジェネレータSOURCERY

1989

■89年1月号
第75部 パズルゲームLAST ONE
第76部 ブロックゲームFLICK
■89年2月号
第77部 高速エディタアセンブラREDA
特別付録 X1版S-OS "SWORD"<再掲載>
■89年3月号
第78部 Z80用浮動小数点演算パッケージSOR
OBAN
■89年4月号
第79部 SLANG用実数演算ライブラリ
■89年5月号
第80部 ソースジェネレータRING
■89年6月号
第81部 超小型コンパイラTTC
■89年7月号
第82部 TTC用パズルゲームTICBAN
■89年8月号
第83部 CP/M用ファイルコンバータ
■89年9月号
第84部 生物進化シミュレーションBUGS
■89年10月号
第85部 小型インタプリタ言語TTI
■89年11月号
第86部 TTI用パズルゲームPUSH BON!
■89年12月号
第87部 SLANG用リダイレクションライブラリDIO.LIB

1990

■90年1月号
第88部 SLANG用ゲームWORM KUN
特別付録 再掲載SLANGコンパイラ
■90年2月号
第89部 超小型コンパイラTTC++
■90年3月号
第90部 超多機能アセンブラOHM-Z80
■90年4月号
第91部 ファジィコンピュータシミュレーションI-MY
■90年5月号
第92部 インタプリタ言語STACK
■90年6月号
第93部 リロケータブルフォーマットの取り決め
第94部 STACK用ゲームSQUASH!
第95部 X68000対応S-OS "SWORD"
特別付録 PC-286対応S-OS "SWORD"
■90年7月号
第96部 リロケータブルアセンブラWZD
■90年8月号
第97部 リンカWLK
■90年9月号
第98部 BILLIARDS
■90年10月号
第99部 ライブラリアンWLB
■90年11月号
第100部 タブコード対応エディタEDC-T
■90年12月号
第101部 STACKコンパイラ

1991

■91年1月号
第102部 ブロックアクションゲームCOLUMNS
■91年2月号
第103部 ダイスゲームKISMET
■91年3月号
第104部 アクションゲームMUD BALLIN'
■91年4月号
第105部 SLANG用カードゲームDOBON
■91年5月号
第106部 実数型コンパイラ言語REAL
■91年6月号
第107部 Small-C処理系の移植
■91年7月号
第108部 REALソースリスト編
■91年8月号
第109部 Small-Cライブラリの移植
■91年9月号
第110部 SLANG用NEWファイル出力ライブラリ
■91年10月号
第111部 Small-C活用講座(初級編)
■91年11月号
第112部 Small-C活用講座(応用編)
第113部 MORTAL
■91年12月号
第114部 Small-C SLANGコンパチ関数

1992

■92年1月号
第115部 LINER
■92年2月号
第116部 シミュレーションゲームPOLANYI
■92年3月号
第117部 カードゲームKLONDIKE
■92年4月号
第118部 オプティマイザO80実践Small-C講座(1)
■92年5月号
第119部 COMMAND.OBJ実践Small-C講座(2)
■92年6月号
第120部 COMMAND.OBJ2実践Small-C講座(3)
■92年7月号
第121部 関数リファレンス実践Small-C講座(4)
■92年8月号
第122部 ワイルドカード実践Small-C講座(5)
第123部 グラフィックライブラリ GRAPH.LIB
■92年9月号
第124部 O-EDIT&MODCNV
■92年10月号
第125部 SLENDER HUL実践Small-C講座(6)
■92年11月号
第126部 EDIT実践Small-C講座(7)
■92年12月号
第127部 MAKE実践Small-C講座(8)

1993

■93年1月号
第128部 EDC-Tの拡張
■93年2月号
第129部 BLACK JACK
■93年3月号
第130部 シューティングゲームコアシステム作成法(1)
■93年4月号
第131部 シューティングゲームコアシステム作成法(2)
■93年5月号
第132部 シューティングゲームコアシステム作成法(3)
■93年6月号
第133部 REVERSI
■93年7月号
特別付録 MSX用S-OS "SWORD"
■93年8月号
第134部 MACINTO-C再掲載
■93年9月号
第135部 7並べ
特別付録 SLANG再々掲載
■93年10月号
第136部 シューティングゲームコアシステム作成法(4)
■93年11月号
第137部 S-OSで学ぶZ80マシン語講座(1)
■93年12月号
第138部 エディタアセンブラREDA再掲載

1994

■94年1月号
第139部 S-OSで学ぶZ80マシン語講座(2)
■94年2月号
第140部 YGCSver.0.20ユーザーマニュアル
第141部 S-OSで学ぶZ80マシン語講座(3)
■94年3月号
第142部 S-OSで学ぶZ80マシン語講座(4)
■94年4月号
第143部 S-OSで学ぶZ80マシン語講座(5)
■94年5月号
第144部 S-OSで学ぶZ80マシン語講座(6)
■94年6月号
第145部 YGCSver.0.30
■94年7月号
第146部 シューティングゲーム作成講座(1)
■94年8月号
第147部 シューティングゲーム作成講座(2)
■94年9月号
第148部 怪しいZ80の使い方(テクニック編)
■94年10月号
第149部 シューティングゲーム作成講座(3)
第150部 怪しいZ80の使い方(未定義命令編)
■94年11月号
第151部 B-GALET2
■94年12月号
第152部 シューティングゲーム作成講座(4)

バックナンバー案内

BACK ISSUES

ここには1994年6月号から1995年5月号までをご紹介しました。現在1994年4～12月号，1995年3～5月号の在庫がございます。バックナンバーはお近くの書店にご注文ください。定期購読の申し込み方法は136ページを参照してください。

1994

6月号
特集　X68000と仲間たち
連載　ハードコア3D/響子 in CGわ～るど/ショートプロ/ローテク工作/ファイル共有の実験と実践/こちらシステムX探偵事務所/ANOTHER CG WORLD
●第5回Oh!Xアンケート分析大会
●新製品紹介　F-Calc for x68k
LIVE in '94　キャミイのテーマ/The End of Love
THE SOFTOUCH　スーパーリアル麻雀PIV/あすか120% BURNING Fest他
全機種共通システム　YGCS ver.0.30

7月号
特集　入門コンピュータミュージック
連載　響子 in CGわ～るど/ショートプロ/ゲーム作りのKNOW HOW/ローテク工作/システムX探偵事務所/マシン語プログラミング/DōGA CGアニメーション講座/ファイル共有の実験と実践
●特別付録　CGA入門キット「GENIE」
●実用講座　Photo CDでカードを作る
LIVE in '94　宇宙刑事ギャバン/究極戦隊ダグダーン/スティング　他
THE SOFTOUCH　麻雀航海記/雀神クエスト/The World of　X68000II 他
全機種共通システム　シューティングゲーム作成講座(1)

8月号
特集　Graphic Movement
連載　響子 in CGわ～るど/ショートプロ/ハードコア3D/ローテク工作/ANOTHER CG WORLD/善バビ/DōGA CGアニメーション講座/石の言葉，言葉の夢
●新製品紹介　X-SIMM VI/Mu-1 GS
SX-WINDOW ver.3.1
LIVE in '94　PURE GREEN/Ridge racer(POWER REMIX)
THE SOFTOUCH　Mr.Do!/Mr.Do! vs UNICORNS/レッスルエンジェルス3
全機種共通システム　シューティングゲーム作成講座(2)

9月号
特集　SX-WINDOW環境セットアップ
連載　響子 in CGわ～るど/ショートプロ/ハードコア3D/ローテク工作/DōGA CGアニメーション講座/善バビ/システムX探偵事務所/ファイル共有の実験と実践
●新製品紹介　X68030 D'ash/MJ-700V2C
●新刊紹介　X680x0 TeX
LIVE in '94　LOVE IS ALL/HELL HOUND/踏切の通過音
THE SOFTOUCH　餓狼伝説SPECIAL
全機種共通システム　怪しいZ80の使い方(テクニック編)

10月号
特別企画　もみじ狩りPRO-68K
連載　響子in CGわ～るど/ショートプロ/ハードコア3D/TeX入門講座/ゲーム作りのKNOW HOW/善バビ/猫とコンピュータ/ファイル共有の実験と実践
●特別付録　もみじ狩りPRO-68K(5″2HD)
●新製品紹介　F-Card V5 for x68k
LIVE in '94　イース2/MSX用GRADIUS2/NATURE
THE SOFTOUCH　スーパーストII/スターラスター　他
全機種共通システム　怪しいZ80の使い方/ゲーム作成講座(3)

11月号
特集　STEP UP BASIC
連載　響子 in CGわ～るど/ショートプロ/ハードコア3D/TeX入門講座/DōGA CGアニメーション講座/システムX探偵事務所/ローテク工作/善バビ
●新製品紹介　BJC-400J/X680x0 Develop. & libc II
Free Software Selection Vol.2
LIVE in '94　ダーク・スペース/ENDLESS RAIN/レナのテーマ
THE SOFTOUCH　スーパーストII/餓狼伝説SPECIAL
全機種共通システム　B-GALETS2

12月号
特別企画　XL/Imageお試し版＋α
連載　響子in CGわ～るど/ショートプロ/ハードコア3D/ファイル共有の実験と実践/DōGA CGアニメーション講座/システムX探偵事務所/ローテク工作/TeX入門講座
●特別付録　XL/Imageお試し版＋α(5″2HD)
●新製品紹介　H.A.R.P/XDTP SX-68K
LIVE in '94　幻想即興曲/きまぐれ オレンジ☆ロード 他
THE SOFTOUCH　魔法大作戦/スーパーストII
全機種共通システム　シューティングゲーム作成講座(4)

1995

1月号(品切れ)
特集　割り切って使うCD-ROM
連載　響子 in CGわ～るど/ショートプロ/ハードコア3D/ファイル共有の実験と実践/DōGA CGアニメーション講座/システムX探偵事務所/ローテク工作/TeX入門講座
●CD-ROMドライブ紹介　CS-CD301X/CDS-E/SCD-200
●新製品紹介　X68000XVI用アクセラレータXellent30
LIVE in '95　ぷよぷよ/ジムノペディNO.1/PRIME
THE SOFTOUCH　バックランド/上海 万里の長城/魔法大作戦/餓狼伝説SP 特別編/スーパーストII 特別編

2月号(品切れ)
特集　MicroProcessingUnit
連載　響子 in CGわ～るど/ショートプロ/ハードコア3D/SX-BASIC公開デバッグ/DōGA CGアニメーション講座/システムX探偵事務所/SX-WINDOWによるDTP
●特別企画　最新ゲーム機を見る
●新製品紹介　Datacalc SX-68K/シャーペンワープロパック
●1994年度GAME OF THE YEARノミネート作品発表
LIVE in '95　サムライスピリッツ/AFTER SCHOOL/白鳥の湖
THE SOFTOUCH　スーパーストII 特別編

3月号
特集　SoundEffects
連載　響子 in CGわ～るど/ショートプロ/ハードコア3D/システムX探偵事務所/ファイル共有の実験と実践/ピコピコエンジン活用講座/SX-WINDOWによるDTP
●SX-WINDOW用ユーティリティ　どっち.X
LIVE in '95　魔法のプリンセスミンキーモモ/別れの曲/ファイナルファンタジーII/宇宙戦艦ヤマト完結編
THE SOFTOUCH　ディグダグ/ディグダグII/VIEW POINT
全機種共通システム　S-OSシステムコールライブラリ

4月号
特集　Let's Play Wonderful GAME
連載　響子 in CGわ～るど/ショートプロ/ハードコア3D/システムX探偵事務所/ファイル共有の実験と実践/DōGA CGアニメーション講座/ローテク工作
●1994年度GAME OF THE YEAR発表
●新製品紹介　TS-6BS1mkII/MJ-5000C/MATIER ver.2.1
LIVE in '95　天聖龍/ファイナルファンタジーVI/ANOTHER DAY/ハートオブザマッドネス
全機種共通システム　S-OSねちねち入門(1)

5月号
特集　Realize Graphic
連載　響子 in CGわ～るど/ショートプロぱ～てぃ/ローテク工作実験室/SX-BASIC公開デバッグ/システムX探偵事務所/ANOTHER CG WORLD
●特別付録　Oh!電脳倶楽部
●新製品紹介　フォント＆ロゴデザインツール
LIVE in '95　ドラゴンセイバー/ミッドナイトレジスタンス 他
THE SOFTOUCH　ボンバーマン ぱにっくボンバー
全機種共通システム　S-OSねちねち入門(2)

ネットサーフィンとクモの巣作り

開かれたインターネット

いまや世界的な規模に発展したネットワークといえるインターネットですが，そこで利用可能なサービスにもいろいろあります。電子メールとニュースは基本です。これらは，NIFTY-Serveなどのパソコン通信からも利用できます。電子メールで連絡を取れる相手がぐんと増えたということで一層欠かせないものとなってきました。

電子メールに関してはパソコン通信とインターネットの垣根はなくなってきたわけですが，その乗り入れはまだまだ限定されたものです。インターネットを自由に活用するには，インターネットに所属している組織に自分が加わるのがいちばんです。

大学，研究所，(ネットワークに理解のある)会社などでしょうか。頭の固いお役所でさえ，少しずつですが，インターネットに取り組む姿勢を見せ始めました。実際，就職先を探す場合でも「職場でインターネットを自由に使えること」という条件を口にする学生も出現してきました。そんな時代です(もっとも，学生側が条件など出せる景気ではないのですが)。

ただ，組織がインターネットに加わっているといっても，一部の人たちだけが利用できるだけで，全員が自由に使えるわけではない場合が多いようなので注意が必要です。大学でも，文系の学部では研究室にいっても使えないところが多そうです。

その点，名古屋大学はかなり画期的で，入学式のときに新入生全員にワークステーション用のユーザーIDが与えられます。理系の学部ではそう珍しくもないと思いますが，全学部生が自由にインターネットを使えるところがいいと思いませんか？

ところで，数的に見れば，インターネットなんて，世の中の多くの人とは関係ない話かもしれませんが，最近はそうでもなくなってきているようです。インターネットプロバイダーとかいう新しい商売ができて，各家庭の電話回線から，インターネットを使うことも可能となってきました。

ですから，パソコンとモデムがあれば誰でもインターネットのサービスを受けることが一応できるわけです。プロバイダーの数も増えて競争も激しいようで，加入料金や利用料金もずいぶんと安くなってきているようです。利用料金としては，月々数千円ぐらいでしょうか。

ただし，インターネットプロバイダーを家庭から個人的に利用することが本当に嬉しいことなのかどうかは，いまのところなんともいえません。

まず，スピードの問題があります。通常の電話回線で9,600bpsとか19,200bps程度だと簡単なメールの交換ぐらいならいいのですが，画像データとかプログラムとかをやり取りすると快適かどうか疑わしいですね。むろん2,400bpsじゃ……。

単にメールを交換するだけならば，ふつうのパソコン通信に加入したほうが，ほかにもいろいろなサービスを受けられるという点で望ましいでしょう。

さらに，月々の利用料金や電話代に見合うほどの面白いサービスを受けられるかどうかという問題があります。これは重要な問題です。しかし，いまはどうであれ，今後，急速に充実していくことは間違いないと思います。止めることのできない大きな流れが起こっているからです。

今回は，現在のインターネットが本当に楽しめるものかどうか判断するための材料をひとつ提供することにしましょう。

世界にまたがるデータベース

最近，いろいろと話題を呼んでいるのが，www(World Wide Web)サービスです。wwwとはインターネット上におけるハイパーテキスト型データベースです。ハイパーテキストとは，断片的なテキストデータの間で自由にリンクを張ることにより表現されている構造を意味し，その例としてはMacintoshのハイパーカードがあります。単にテキストだけではなく，ハイパーカードのようにグラフィックや音声などもデータとなる場合にはハイパーメディアとも呼びます。

たとえば，ある文章を書いているときに難しい専門用語が出てきて，その説明が必要だとします。その場合，ハイパーテキストでは，説明を同じ文章中に挿入するのではなく，別のデータとして，説明文をそこからリンクすることができます。

僕がいま書いているこの文章でも(このような紙の媒体ではハイパーテキスト化は困難ですが)，ハイパーテキスト化すれば，もっとわかりやすく書くことができます。

たとえば，「ハイパーテキスト」という言葉の簡単な説明を書いたわけですが，この言葉が最初に出てきたところにリンクを張り，説明の部分を別のテキストにするわけです。

そうすれば，文章の構造がよくわかるようになりますし，「ハイパーテキストなんて知っているさ」という人は無駄な説明を読まなくてすみます。さらに，僕があとでハイパーテキストの説明を修正する場合にも，その部分だけをいじればよくなります。

wwwでは，ハイパーカードと同様に，音声や画像などのデータも自由に組み合わせることができます。マルチメディア対応とでもいうところでしょう。さらに，wwwの大きな特長はネットワーク対応型であることです。要するに，リンクが世界的な規模で網の目のように張りめぐらされているのです。wwwという名前そのものですね。

インターネットに接続された別のマシンに入っているデータも同様にリンクできます。あるデータを1つのウィンドウで見ているときに，マウスである部分をクリックすると，国をまたいだ別の計算機の中のデータが表示されることもあります。しかも，アクセスしている側はそれを特別に意識しないのです。

ネットサーフィン

インターネット上のwwwの情報を自由気ままにたどって遊ぶことを「ネットサーフィン」と呼びます。あちらの人はネーミングもナイスですね。クモの巣(web)の上を這い回るのはクモに決まっているのにサーフするのですからね。

サーフィンするにはそれ用のプログラム(ブラウザと呼ぶ)が必要です。昔はMOSAIC(モザイク)というのが有名でしたが，いまはNetScape(ネットスケープ)がベストのようにいわれます。最大の魅力はスピードでして，NetScapeはMOSAICに比べてかなり速いといわれています。

NetScapeにはワークステーション版やパソコン版がありますが，教育機関などでの使用は無料です。毎週のようにバージョンアップされていますが，それもNetScape社のwwwサーバから簡単に取り寄せることができます(国内のミラーサイトから取り寄せるのが礼儀ですが)。

どんなブラウザを使おうとも面白いデータをみつけられなければ意味がありません。しかし簡単にみつける方法はありません。なぜならば，インターネットに接続されている世界各地の無数の計算機上で，いまこの瞬間も面白いデータが書かれつつあるのですが，そのデータすべてが集中的に管理されているわけではないからです。

パソコン通信のような真ん中にデンと大型コンピュータがあって，という形式ではなく，対等な計算機が各地に分散していて，それを結んでいるという形をとるネットワークでは，基本的にそのような集中的な管理はそぐわないからでもあります。

しかし，だからこそ，サーフィンの楽しさが生まれるわけです。自分だけの素敵な波をうまくみつけたという秘かな楽しみが。しかも，皆に「ここはいけるぞ」と公開するとアクセスが殺到して，軽快に波乗りできなくなってしまうのです。

とはいうものの，ネットワーク上でも電話帳や地図に相当するものはあります。たとえば，キーワードを入れるとそれに関係するデータを見せてくれるサービスや，日本地図や世界地図が画面に表示され，各地のサーバをグラフィカルに検索できるサービスなどです。

NTTのwwwサーバにアクセスすると日本にあるサーバを地図を選びながら探すことができます。たとえば，ある大学の研究室を探し出したとします。www用ページをもっている研究室も増えてきました。好きな学生が配属されると作るのでしょうね。

典型的なパターンとしては，最初のページで，研究室の名前がどんと上に出て，その下にデータ項目が並びます。研究プロジェクトや研究室構成メンバーなどです。研究室構成メンバーという項目名をマウスでクリックすると(リンクがあるところは色が変わっており，そのページを見終えるとまた別の色に変わっています)，メンバーの一覧表が並ぶページに移ります。

さらに，そのメンバーの名前のところをクリックすると，今度は各メンバーのプライベートな情報を見ることができます。一般的には，顔写真，略歴，趣味，あるいは，自分が気に入った国内外のページへのリンクなどです。飼っている猫の写真や嫁さん自慢の写真，自分の演奏したギターの音などもありました。でも，あまり見回っていないこともあって，これはというのは少ないですね。

最近は，インターネットの可能性に注目した会社や団体がwwwのページを作るようになってきました。ソニーも作っていて，この間はMDに関する情報を調べました。ほかでは，朝日新聞が新しいメディアを作る準備を始めていて，その新しいメディアの編集長の挨拶などが載っていました。人工生命に関する情報を提供しているところもあります。クレジットカードの番号を入力するだけで通信販売も利用できますし，書籍のデータベースもあります。

「ネットサーフィン」をやっていると思いもかけない情報に出会えます。僕も経験不足ながら，いろいろ面白いページに出会いましたが，秘密にしておきましょう。「本物のサーファーは格好のポイントを決してしゃべらないのだっ！」(本当かな？)。

情報発信は簡単だ

自分を紹介するページを作ってみることにしました。表示させたいデータをhtml（Hyper Text Markup Language）という言語を使って記述するのです。言語といってもそう難しいものではありません。たとえば，次のように書けばよいのです。

```
<HEAD>
<TITLE>ホームページの作り方</
TITLE>
</HEAD>
<BODY>
<H1>はじめに</H1>
wwwでホームページを書くことは難
しいことではありません。<P>
……
</BODY>
```

<HEAD>と</HEAD>で囲まれたヘッダ部と，<BODY>と</BODY>で囲まれた本体で構成されます。ヘッダ部には，<TITLE>と</TITLE>で囲まれたページタイトルを書きます。このタイトルは多くのブラウザでウィンドウタイトルの部分に表示されます。<H1>と</H1>で囲まれた部分は，レベル１の見出しを意味します。レベル２，レベル３になるほど見出しのレベルは低くなります。通常，見出しのレベルは表示されるフォントの大きさに対応します。<P>は段落の終わりを意味し，改行されます。

なお，<HEAD>……</HEAD>や<BODY>……</BODY>は省略されることがしばしばあります。したがって，単にテキストを表示するだけならば，ウィンドウのタイトルを指定するとか，見出しを指定するぐらいで，すでにある通常のテキストを簡単にhtml形式に変換して表示できるのです。

ただし，いくつか注意すべき点があります。

注意１　htmlは文章の構造を明示する

ワープロでは，改行したり，１，２，３とふって箇条書きする場合，自分で番号をふったり字下げ(インデント)などしたりして形を整えます。しかし，hmtlでは，箇条書きするようなリスト構造は，たとえば，

<UL><LI>項目１<LI>項目２……</UL>

のようにして構造を明示します。

それにより，ブラウザがインデントしたり，改行したりを自動的にやってくれるのです。また，逆に，改行や空白を多数入れても無視されますので注意しましょう。

注意２　ブラウザの裁量にまかせる

文字列を<EM>……</EM>，<STRONG>……</STRONG>，<CITE>……</CITE>などではさむと，文字に対して，それぞれ，強調，さらなる強調，引用などのスタイルが指定されます。これらは，論理的スタイルといい，その文字列をどのように画面に表示するかはそれぞれのブラウザにまかせられています。多くの場合，それぞれ，斜体，太字，斜体で表示されます。したがって，最初から，斜体の指定<I>……</I>，あるいは，太字の指定<B>……</B>のような物理的スタイルを指定しても同じことになります。

なぜ，論理的スタイルがわざわざ用意されているかというと，どんなブラウザ，あるいは計算機でも対応できるようにということでしょう。たまたま現在では強調は斜体ということになっている機械が多いだけであって，もっと詳細な区別が用意される処理形ならばそれを使えるのです。そういう意味で将来の発展までカバーしているともいえます。また，太字や斜体が画面上に表示できない計算機でもその機種なりに，色を変えたり，下線を引いたり，好きに設

ネットサーフィンとクモの巣作り

定できるわけです。

ですから，物理的な指定で限定するよりは，特定の意味づけを指定するという論理的スタイルを使うほうが望ましいといえましょう。

肝心のハイパーテキスト機能ですが，特に難しいことはありません。たとえば，

<A HREF = "ファイル名">……</A>

とすれば，この間の文字列をマウスでクリックすると，ファイル名で示されるページに飛ぶのです。ファイル名の書式において，インターネット上の別の計算機のファイルを指定できるというのがミソです。

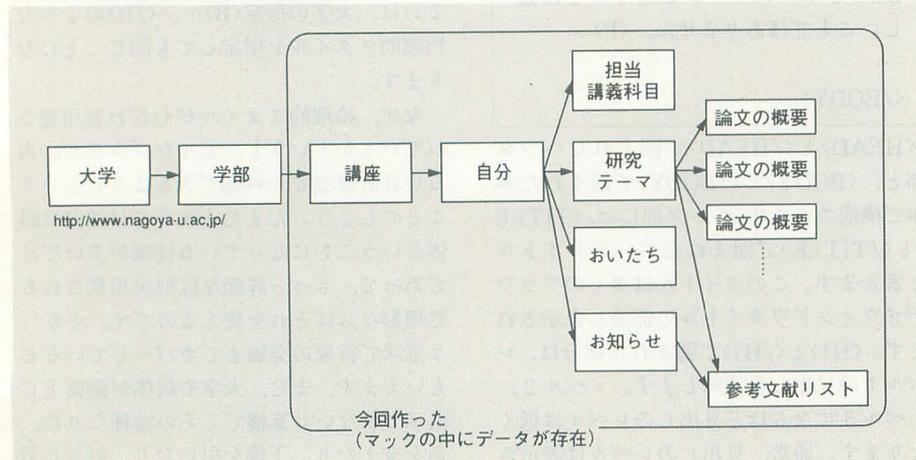
一方，グラフィクのデータを画面に表示するのも簡単です。たとえば

<IMG SRC="グラフィックファイル名">

とすればいいのです。横線を引くだけならば，<HR>一発です。

ここまでで述べてきたことがhtmlの基本的な説明で，さらにいくつかの拡張機能をマスターすれば，あとは好き勝手にやればいいのです，グラフィックを張りつけたり，音を入れたりして。しかし，ここでもうひとつ大切な注意があります。

注意3　無駄に通信量を増やさない

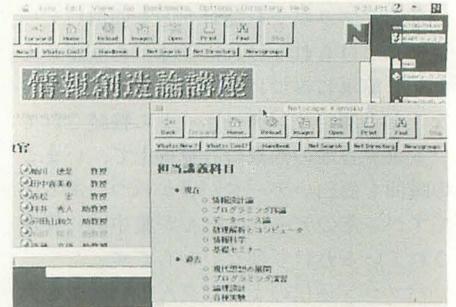
ネットワークは公共の道路ですから，道路を混雑させないように注意する必要があります。なるべく無駄なデータは流さないようにするのです。文字データならどうせタカがしれていますが，グラフィックに凝るといくらでもデータの量を増やしてしまいます。それに，ウィンドウ1つ表示するのに長い時間待たせるようなページは2度と見てくれなくなります。

また，遅い回線でアクセスしている人はブラウザでグラフィックデータは読み込まないという設定にしている人も多いようなので，力をかけすぎても意味がないということになります。

データ量を抑えつつも視覚効果を上げること，これはマルチメディアとインターネットの融合において重要なことであり，そこにはセンスが要求されてくるのです。

自己紹介は苦手ですが……

知ったかぶりして，htmlについて書いてきましたが，あまり真面目に学んだことはありません。というのも，NetScapeなどのブラウザでは表示したページのhtml言語による記述をすべて見ることができるのです。ですから，実例は豊富にあるので，いくらでもそれを真似することができるわけです。もちろん，まっとうに勉強するためのhtml入門のページもきちんとインターネット上に用意されています。

自己紹介のページをえいやっと作りました。大学はもちろんwwwのページを用意しており，情報文化学部もまた各講座名を並べたページを用意してあります。そこで，まず，自分の所属している情報創造論講座のページを作りました。

まず，サーバをどうするかを考えないといけません。サーバプログラムとは，見たいという要求がほかからきたときにそれを受け付け，相手にそのページのデータを送るという働きをするものですが，Macintosh用にもある(MacHTTP)ので，それを利用することにしました。

作成したページ

講座のページは，単に，ここに所属している先生の名前を羅列しただけです。ただ，「情報創造論講座」というタイトルはただ大きく表示するのではあんまりですから，PhotoShopというソフトで，その文字を加工しました。昔，学校で金属プレートを釘でとんとんと打って模様を浮き彫りにしたようなイメージです。

次に，自分のページですが，顔写真にビデオで取り込んだ，やや横を向いたデータを使うことにしました。それから，研究テーマ，簡単な生い立ち，お知らせなどの項目を書き，そこをクリックすると，それぞれのページに飛ぶようにしました。

一応，項目だけ作って，リンクは張らずにそのままにして帰宅し，次の日，来てみると，見られていました。自分の大学内からが主ですが，別の大学とか，あるいは，アメリカからとか。となると，やはり，もう少し真面目に書かねばということで，半日をつぶして書きました。

ページを英語と日本語の両方で書くことにしました。上のほうにEnglishとJapaneseというリンクを用意しておいて，Englishをクリックすれば，英訳したページを見ることができるようにしたのです。

研究プロジェクトのほうも，各テーマについて1つずつ参考となる自分の論文名を挙げました。さらに，もっと興味のある人には，クリックするとその論文の概要も見られるようにしました。

とりあえず，このぐらいで放ってあります。いやー，まだまだお粗末なもので，もっと面白いものを作りたいとは思いますが，のめり込みそうなので自粛しています。まあ，おいおい体裁を整えていくことにしましょう。遠い将来，とっておきのページができたらお知らせします。

名古屋大学のwwwサービスの一部

(で)のショートプロぱーてぃ――その69

環境乱舞なENVRANDだ

Komura Satoshi 古村 聡

今月は，起動ごとランダムに環境変数を書き換えてくれるENVRAND.C，百人一首練習プログラム，そしてSX-WINDOW用のリモコンが登場です。この季節にふさわしく家でじっくり使えるプログラムが4本。活用してね。

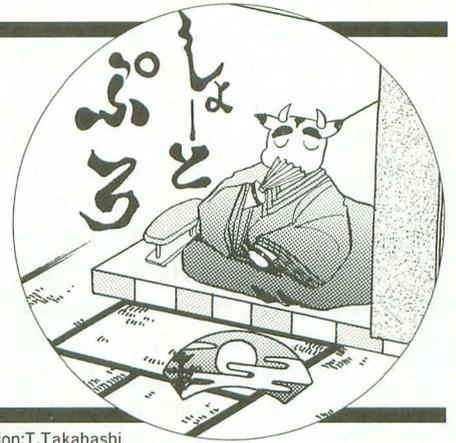



illustration:T.Takahashi

やー，驚いた。なにが驚いたかというと，木村太郎さんのテレビ番組での発言でありますよ。この号が出る頃にはもう解決していてほしいと切に願う私なんでありますが，この原稿を書いている4月15日の日にオウム真理教がなにかするかもしれないっていう話があったんですよ。彼はその前日のテレビのニュース番組で「パソコン通信がこのデマを広めている感がある。こんなときにはパソコン通信は自粛すべきだ」ってのたもうたのであります。

常日頃，報道の自由を標榜しているマスコミが，個人の表現の自由を制限するような発言をすることも驚きなんですが，それよりも驚いたのはパソコン通信がテレビのニュース番組で登場するほど，一般的なモノになったということであります。1200bpsのモデムやカプラで通信していた，まだパソコン通信専門誌はおろか，パソコン誌でもほとんど話題にならなかった時代から，まだ10年もたってないんですからね。

もちろん，パソコン通信が誰でも発言できるメディアで，歴史も浅くチェックの機構もまだないため，そのなかには相当眉唾モノも含まれていることは認めます。しかし，考えてみればテレビだって新聞だって全部が全部本当のことを書いているなんて保証はありません。そうしている努力は認めるけど努力が常に正しい結果を生み出すとは限らないわけです。情報というのはそもそも受け取る側が真偽を判断して選択しなくては，危なくて使えないものだとも思えるのであります。ということは発言するのも自由，デマを信じるも信じないも自由，というパソコン通信の姿こそ，情報のもともとあるべき姿ではないかと私は思うのであります。

その意味でいえば，むしろ見る側の選択する努力をスポイルしようとする木村氏の発言こそ，大きなお世話ではないかと私には思えるのでありますが。いかがなものなんでありましょうかね。

環境変数が適当～なのだ

さてさて。どういうわけなんだか，X68000って電源入ったとたんにPICなグラフィックを表示したり，AUTOEXEC.BATで音楽を流してみたりしたくなるんですよね。かくいう私のX68000も起動と同時に久川綾嬢のサンプリングボイスで「おはようございま～す」とかいってくれちゃったりするんだ，これが。

でもでもこういうのって最初は楽しいんですけど，毎回同じ絵とか同じセリフだとそのうち飽きてきちゃうんですよね。そんな悩みを一気に解消してくれちゃうのが，今月の1本目のプログラムです。ランダムに環境変数を発生させるENVRAND.X，どうぞっ!

★ENVRAND.X for X680x0

(要Cコンパイラ)

奈良県 寺田 智

このプログラムは環境変数を設定ファイルに書かれた候補のなかから選んで設定してくれるプログラムです。

このプログラムはCコンパイラver.2.0以上用のソースリストとして掲載されています。リスト1をエディタから入力してセーブしてから，

A>cc /Y ENVRAND.C

でコンパイルして，実行ファイルENVRAND.Xを作ってください。ちなみにこのプログラムをコンパイルするとWarningは3つ出ます(実行に支障はありません)。

さて，実行ファイルができたら，プログラムの使い方です。このプログラムは選択候補のなかからランダムに設定値を選択し，環境変数に設定します。

A>ENVRAND [設定ファイル名]

これで環境変数を設定ファイルにあげられた候補のなかからランダムに設定してくれます。設定ファイルは次のような形になっています。

・環境変数は<>でくくって行頭に表記。以下選択肢を一気に続ける
・選択肢の最後は改行でしめくくる
・あとはどこにコメントを書いてもいい

たとえば，下の例の“STARTPIC”と“運勢”が環境変数として設定されます

●(設定ファイルの例)

```
<STARTPIC> ←設定する環境変数名を
<>でくくって表記します
c:¥GRPS¥GRP01.PIC ↓以下，選択される
c:¥GRPS¥GRP02.PIC ↓文字列を続けます
c:¥GRPS¥GRP03.PIC ↓ファイル名の後ろ
c:¥GRPS¥GRP04.PIC ↓はなにを書いても
c:¥GRPS¥GRP05.PIC ↓かまいません
c:¥GRPS¥GRP06.PIC  (注釈にどうぞ)
c:¥GRPS¥GRP07.PIC
<運勢> ←次の環境変数
大吉。いいことがあるようです。
中吉。よろしいんじゃないでしょうか。
小吉。可もなく不可もなく。
吉。平凡な1日ですね。
凶。気をつけましょう。
```

で，たとえば，毎回異なったグラフィックを起動したいのであれば，設定ファイルに画像データを選択肢として設定し，AUTOEXEC.BATに，

ENVRAND〔設定ファイル〕
HAPIC %%STARTPIC%%

と書けば，設定ファイルのSTARTPICの候補のなかから絵を探し出して表示してくれます。

また，コマンドラインで，

A>ENVRAND /?

とすると簡単な使用方法が表示されます。

いや～，待ってました，こういうプログラムっ。私もAUTOEXEC.BATで出してる声にそろそろ飽きてきて，どうしようかと思っていたんですよね。だもんで，ランダムにPCMを再生するプログラムを作ろ

うかとたくらんでいたんですが……
そうかこういうエレガントな方法もあったわけだ。寺田さん偉いっ。

おかげで私のマシンは，サンプリング音声を50近くハードディスクに登録してあって，毎回違うセリフをいってくれます。でもって，ときには動画再生してくれちゃったりもしてます（envrandをうまく２重に使ってるんですね)。ぜーんぶ，セラムンの亜美ちゃんですけどねん。なかには「きゃあっ！」なんてのも入ってますけど，ホッホホホ☆（ってあたしゃ危ない人かい)。

……しかし，これだけで容量100Mバイトのハードディスクの20Mバイト以上を使ってしまってるんですけど。趣味とはいえ困ったもんだ。

実用的かも百人一首

続いてのプログラムは，実用プログラム（なのかなあ)，X-BASIC用の百人一首プログラムです。どうぞっ。

百人一首.BAS for X680x0
（X-BASIC，要XSPRITE.FNC，XVI.FNC）
佐賀県　服部　誠

いわずと知れた百人一首です。このプログラムを使うにはテキストファイルの句のデータと，PCMファイルが必要となります。まず，BASICのプログラムファイルであるリスト２を打ち込んでセーブしてから，リスト３の句データをエディタで打ち込んで，“いっしゅ.DAT”という名前で保存してください。PCMファイルはファイルサイズが320バイトのものを２つ使います。ひとつが正解用で，もうひとつが不正解用です。これは皆さんのお好みで使用してください。ちなみに投稿プログラムでは，('91年５月号）に付録でついてきたSIONⅡのPCMファイル「BD1.PCM」「CHH.PCM」が使用されていました。

さて，プログラムの入力が終わったら今度は遊び方。タイトル画面で，CONFIGを選ぶと，上の句，下の句について，表示方法を変更することができます。ただし，上の句については決まり字まで，上の句のみを選ぶと下の句が出ません。ですから，下の句がわからないときは，調べるか片っ端から選んでください。

STARTを選ぶと，ゲームが始まります。左クリックで札を取り，右クリックで50枚ずつ切り替えます。正しい札を取るまで，次の歌へは進むことができません。100枚取り終えると，成績が出ます。

難しいのは，下の句をいかにも本物の札のように表示しているところです。現在は100枚の札を50枚ずつ出してそれを交互に表示させています。

うーみゅ，これはわからんっ！　百人一首って上の句を聞いて下の句が書いてある札を取るんですよね～。１句も知らない私は全然遊びようがないのでありました。「犬も歩けば」「棒にあたる」とかなら知ってるんだけど（カルタじゃないんだっつの)。まじめに覚える気があれば，練習には役立つと思うんですけどね。投稿原稿によれば，このプログラムを制作した服部さん，実際に高校の百人一首大会の練習のために使い，なんと優勝してしまったんだとか。ちょっとすごい実用ソフトかもしれない。これで一から覚えれば私でも本当にうまくなれるんかしら……。

もし，これ使って上達したって方，いたら教えてくださいまし。

百人一首.BAS

パラサイトなプログラムなのだ

さて，今月最後のプログラムはSX-BASIC用のプログラムですね。SX-BASICでテレビをコントロールするプログラム，画面設定.SXBと，りもこん.SXBです。

画面設定.SXB
りもこん.SXB for SX-WINDOW
（要SX-BASIC，SAdjust.r）
千葉県　中上　匠

このプログラムは福嶋氏作のSAdjust.rを利用して，SX-WINDOW上で画面モードを変更するプログラムと，SX-WINDOW用テレビリモコンプログラムです。

画面設定.SXB＆りもこん.SXB

リスト４，５をメモ帳などから打ち込んでセーブしたら，使い方。

まず，SAdjust.rを組み込んでおいてください。それからSX-WINDOWを起動します。続いて，SX-BASICのウィンドウに画面設定のアイコンを放り込むと，画面設定のウィンドウが開きます。好きな画面モードのボタンを押せば，設定が切り替わります。1024×768ドットモードは標準ディスプレイでは映りません。注意してください

そうそう，画面モードですが，

・768×512　　　標準のモード
・768×576　　　お勧めのモード。アスペクト比が１：１，画面の比率が4:3なので普通のディスプレイにぴったり
・512×512　　　SX風船用のモード
・512×512テレビ　スーパーインポーズ
・640×480　　　VGAもどき。すごく狭い
・1024×768　　ハイレゾ。Oh!X94年6月号参照

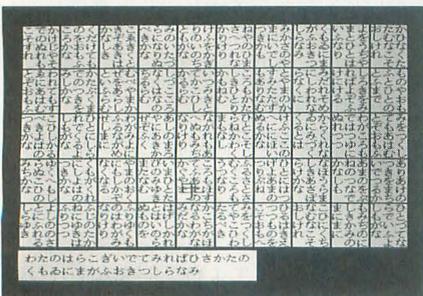
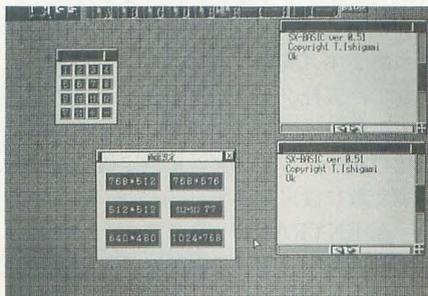
となっております。ちなみに512×512テレビでは，背景を16×16ドットの透明にしておくのが超オススメでございます。なんつっても，これでコンパイルしながらテレビが見られる!(しかもCPUパワーは損なわれない)

それから，続いてリモコンの使い方ですね。せっかく画面設定でテレビが見られるんだから，やっぱりリモコンも作らなきゃ（え？　キーボードコントロール？　忘れましょう，うちX68000CompactXVIだし)。

SX-BASICのウィンドウに「りもこん」のアイコンを放り込むと，リモコンウィンドウが開きます。“V”はビデオ/テレビ，“SI”はスーパーインポーズのON/OFF，“＋”，“－”は音量調整です。まず，画面設定.SXBでテレビモードにしてからお使いください。

うーむ。このプログラムって実際に動く部分はSAdjust.rなわけだけど，SX-BASICで動かしてみるとちょっと違って見えるから不思議ですね。こういうままでOh!Xで掲載されたプログラムを利用するプログラムも歓迎しますよ。特にSX-WINDOWで使えるプログラムはまだ少ないですから，SX-BASICのものなんかいいかもしれないですね（作者の石上君は海外に連行されちゃってるみたいだけど)。

さて，今月はこれにて。

今年はなんだか年の始めから暗いできごとが続いてますね。来月こそはなにもない，よい月でありますように切に願う私であったりします。それではまた来月お会いしましょう。

リスト1 ENVRAND.C

```
  1: /*
  2: *  艶舞楽園
  3: *                                  envrand.c Ver 1.03
  4: */
  5:
  6: #include        <stdio.h>
  7: #include        <doslib.h>
  8: #include        <stdlib.h>
  9:
 10: #define         ERVER           "1.03"
 11: #define         ERDATE          "1995.3.15 "
 12: #define         FAIL            -1
 13: #define         BLOCK_END        1
 14: #define         FILE_MAX        50
 15: #define         LINE_MAX        80
 16: #define         ENVSTART        '<'
 17: #define         ENVEND          '>'
 18: #define         CR              '¥n'
 19: #define         SPC             ' '
 20: #define         TAB             '¥t'
 21: #define         WAIT_TIME       20000
 22: #define         FILE_NONE       1
 23: #define         ENVELOPE        2
 24:
 25: int     lineget(work);
 26: void    err(errnum);
 27: void    usage();
 28: void waittime(count);
 29:
 30: int     i;
 31: FILE    *fp;
 32:
 33: void main(argc,argv)
 34: int argc;
 35: char **argv[];
 36: {
 37: unsigned char   *env=0;
 38: unsigned char   *file;
 39: unsigned char   *path;
 40: unsigned char   envfile[50];
 41: unsigned char   envelope[LINE_MAX+1];
 42: unsigned char   filename[FILE_MAX][LINE_MAX+1];
 43: struct PDBADR   *pdbadr;
 44:
 45: int             fchar , result , fileno , select;
 46:
 47: /* Main START */
 48:     file = (argc > 1) ? argv[1] : "envrand.fil";
 49:     if (!strcmp(file,"/?")) usage();
 50:     printf("《 艶舞楽園 envrand.x Ver %s by qoop.》¥n",ERVER);
 51: /* 設定ファイルの作成 */
 52:     pdbadr = GETPDB();
 53:     path = &(pdbadr->exe_path[0]);
 54:     envfile[0] = NULL;
 55:     strcat(envfile , path);
 56:     strcat(envfile , file);
 57: /* File Open */
 58:     srand((unsigned int)GETTIM2());
 59:     if ((fp = fopen( envfile , "r")) == NULL) err(FILE_NONE);
 60: /* eofまで繰り返し */
 61:     while (feof(fp) == NULL) {
 62:      if ((fchar = fgetc(fp)) == ENVSTART) {
 63:       if (fchar == EOF) break;                      /* EOFなら終了
*/
 64:       lineget(envelope);
 65:       for (fileno = 0 ; (((result = lineget(filename[fileno]))
 != FAIL)
 66:               && (result != BLOCK_END) && (fileno <= FILE_MAX)) ; fi
leno++);
 67:               select = rand() % fileno;
 68:               if (SETENV(envelope,env,filename[select]) < 0) err(ENV
ELOPE);
 69:               printf("  環境変数%sに%sを設定しました。¥n",envelope,f
ilename[select]);
 70:       }
 71:       else  lineget(envelope);      /* 空読み */
 72:       }                             /* 処理終了 */
 73:       fclose(fp);
 74: }
 75:
 76: int lineget(work)
 77: char    work[LINE_MAX];
 78: {
 79:       if (fgets(work,LINE_MAX,fp) == NULL) return(FAIL);  /* 壱
行読み込み */
 80:       if (work[0] == CR) return(BLOCK_END);
 81:       for (i=0 ; ((work[i] != ENVEND) && (work[i] != SPC)
 82:               && (work[i] != CR) && (work[i] != TAB)) ; i++);
 83:       work[i] = NULL;
 84: }
 85:
 86: void    err(errnum)
 87: int errnum;
 88: {
 89:       switch (errnum) {
 90:        case FILE_NONE:
 91:        fprintf(stderr,"設定ファイルがありません！！¥n");
 92:        break;
 93:       case ENVELOPE:
 94:        fprintf(stderr,"環境変数への設定を失敗しました。¥n");
 95:        break;
 96:       }
 97:       exit(FAIL);
 98: }
 99:
100: void    usage()
101: {
102: int             a,b;
103: unsigned char   qoop[50];
104:       strcpy(qoop,"                           by qoop.¥0");
105:       printf("¥n《 艶舞楽園 envrand.x Ver %s 》¥n",ERVER);
106:       printf("-----------------------------------¥n");
107:       printf(" 勝手に環境変数を設定します。      ¥n");
108:       printf(" 使い方:envrand [設定ファイル名] ¥n");
109:       printf("   [設定ファイル]                ¥n");
110:       printf("   <環境変数名>       ┐このような ¥n");
111:       printf("    設定する候補1     │ファイルを ¥n");
112:       printf("    設定する候補2     ┘作成します。¥n");
113:       for (a=25;a>=0;a--){
114:           printf("%s¥r",&qoop[a]);
115:           for (b=0;b<(WAIT_TIME/5);b++);
116:       }
117:       for (a=0;a<3;a++){
118:           printf("                %s  by q--p.¥r",ERDATE);
119:           for (b=0;b<WAIT_TIME;b++);
120:           printf("                %s  by qoop.¥r",ERDATE);
121:           for (b=0;b<WAIT_TIME;b++);
122:       }
123:       printf("¥n¥n");
124:       exit(FAIL);
125: }
```

リスト2 百人一首.BAS

```
 10 /*
 20 /*  百人一首.BAS
 30 /*
 40 /*   Programed by Hattori.Makoto
 50 /*
 60 /*          要 XSPRITE.FNC
 70 /*
 80 randomize(val(right$(time$,2))*200)
 90 screen 1,1,1,1:console ,,0:window(0,0,511,511)
100 dim int Card_no(1,100),Card_Len(1,100),Mode(1),Field(1,9,4)
110 dim str Card_Mas(1,100)[50],start_time
120 char SEIKAI(319),BATU(319)
130 int Nbr,end_flg,fal,rig,bonus,hi_score
140 Mode(1)=2:Mode(0)=3
150 Sprite_set():img_scrn(1,1,1):vpage(15):sp_disp(1):sp_on()
160 Read_Data():PCM()
170 while 1
180   end_flg=title():if end_flg=-1 then break
190   mix()
200   Card_Draw_Shimo()
210   main()
220   ending(rig,fal,bonus)
230 endwhile
240 end
250 func main()
260 int bl,br,x,y,zx,zy,cx,cy,pg=1,no=1,i,Cno,count=1,flg1=0,flg2=0
270 str temp
280 mouse(4):msarea(6,1,505,410):vpage(5):apage(2)
290 sp_disp(1):sp_on()
300 box(6,415,310,472,8):fill(5,416,309,471,15)
310 rig=0:fal=0:bonus=0:start_time=time$
320 while count<>101
330   msstat(x,y,bl,br)
340   mspos(x,y):sp_loc(0,x,y)
350   i=i+1:Cno=Card_no(0,count)
360   if Nbr<=17 and i=25 then Card_Draw_Kami(Cno):i=0
370   if Mode(0)=3 and Nbr=18 and i=100 then Card_Draw_Kami2(Cno)
380   if bl=-1 and flg2=0 then flg2=1
390   if bl=0 and flg2=1 then {
400     mspos(zx,zy):cx=(zx-5)/50:cy=zy/82
410     if Field(pg-1,cx,cy)=Card_no(0,count) then {
420       Match(cx,cy,pg-1):count=count+1:flg2=0
430       rig=rig+1:i=0:bonus=bonus+(17-Nbr):Nbr=0 }
440       else a_play(BATU,4,3):flg2=0:fal=fal+1
450   }
460   if br=-1 and flg1=0 then flg1=1
470   if br=0 and flg1=1 then {
480     flg1=0:pg=pg+1:if pg=3 then pg=1
490     vpage(pg+4)}
500 endwhile
510 sp_off():for i=0 to 3:apage(i):wipe():next
520 endfunc
530 func ending(rig,fal,bonus)
540 str end_time,temp:end_time=time$
550 int t1,t2,score,fp,x,y,bl,br
560 t1=val(left$(start_time,2))*3600+val(mid$(start_time,4,2))*60+va
l(right$(start_time,2))
570 t2=val(left$(end_time,2))*3600+val(mid$(end_time,4,2))*60+val(ri
ght$(end_time,2))
580 img_scrn(0,1,1):score=(6000-(t2-t2))-fal+bonus
590 locate 12,1:color 5:print"成績発表":color 3
600 locate  7,4:print using "時間         ####秒";t2-t1
610 locate  7,6:print using "総クリック ####回";fal+rig
620 locate  7,8:print using "誤クリック ####回";fal
630 locate 7,10:print using "思賞         ####点";bonus
640 locate 7,13:print using "総得点       ####点";score
650 if hi_score<=score then {
660 locate 12,15:color 6:print"最高得点":color 3
670 fp=fopen("SCORE101.DAT","W")
680 temp=str$(score):fwrites(temp,fp)
690 } else {
700 locate 7,15:print using "最高得点   ####点";hi_score}
710 while bl=0:msstat(x,y,bl,br):endwhile:cls
720 endfunc
730 func Match(cx;int,cy;int,pg;int)
740 a_play(SEIKAI,4,3)
750 fill(7,416,309,471,15):vwait(15):cls
760 apage(pg)
770 fill(cx*49+cx+6,cy*81+cy+1,(cx+1)*50+5,(cy+1)*82,0):apage(2)
780 endfunc
790 func mix()
800 int i,j,temp,rbox
810 for j=0 to 1
820   for i=1 to 100
830     Card_no(j,i)=i
840   next
850 next
860 for j=0 to 1
870   for i=1 to 100
880     rbox=0:temp=Card_no(j,i)
890     while rbox=0:rbox=int(rnd()*100):endwhile
900     Card_no(j,i)=Card_no(j,rbox):Card_no(j,rbox)=temp
910   next
```

```
 920 next
 930 endfunc
 940 func Card_Draw_Kami(Cno;int)
 950 int col
 960 str temp
 970 temp=mid$(Card_Mas(0,Cno),Nbr*2+1,2)
 980 if Mode(0)=0 and Nbr<Card_Len(0,Cno) then col=5 else col=1
 990 if Mode(0)=1 and Nbr>=Card_Len(0,Cno) then return()
1000 symbol(Nbr*16+14,423,temp,1,1,1,col,0):Nbr=Nbr+1
1010 endfunc
1020 func Card_Draw_Kami2(Cno;int)
1030 symbol(14,446,Card_Mas(1,Cno),1,1,1,1,0)
1040 endfunc
1050 func Card_Draw_Shimo()
1060 int x,y,k,i,j,no,count,flg=0,col,Cno
1070 str temp[50]
1080 dim int gdata(4099)
1090 apage(1):box(6,1,55,82,8):fill(7,2,54,81,15):get(6,1,55,82,gdata
)
1100 for k=0 to 1
1110 apage(1-k)
1120   for x=0 to 9
1130     for y=0 to 4
1140       put(x*49+x+6,y*81+y+1,(x+1)*50+5,(y+1)*82,gdata)
1150       no=no+1:flg=0:count=0:Field(1-k,x,y)=Card_no(1,no)
1160       for j=1 to 3
1170         for i=0 to 4
1180           count=count+1
1190           Cno=Card_no(1,no)
1200           temp=mid$(Card_Mas(1,Cno),(j-1)*10+2*i+1,2)
1210           if Mode(1)=0 and count<=Card_Len(1,Cno) then col=5 els
e col=1
1220           symbol(((x+1)*50)-16*j+5,(y*81+y+1+1)+16*i,temp,1,1,1,
col,0)
1230           if Mode(1)=1 and count=Card_Len(1,Cno) then flg=1:brea
k
1240         next
1250         if flg=1 then break
1260       next
1270     next
1280   next
1290 next
1300 endfunc
1310 func Read_Data()
1320 int fp,i,fp2
1330 str temp[50]
1340 fp=fopen("いっしゅ.DAT","r")
1350 for i=1 to 100
1360   Card_Len(0,i)=fgetc(fp)-48
1370   freads(temp,fp):Card_Mas(0,i)=temp
1380   Card_Len(1,i)=fgetc(fp)-48
1390   freads(temp,fp):Card_Mas(1,i)=temp
1400 next
1410 fclose(fp)
1420 error off:fp=fopen("SCORE101.DAT","R"):error on
1430 if fp=-1 then {fp2=fopen("SCORE101.DAT","C")
1440 temp="0":fwrites(temp,fp2)
1450 fclose(fp2):hi_score=0} else freads(temp,fp):hi_score=val(temp)
1460 endfunc
1470 func Sprite_set()
1480 char data(255)
1490 int i,j
1500 img_scrn(1,2,1):vpage(0):sp_init():sp_xinit():sp_clr()
1510 symbol(0,0,"手",2,2,1,3,0)
1520 get( 0, 0,15,15,data):sp_def(0,data)
1530 get(16, 0,31,15,data):sp_def(1,data)
1540 get( 0,16,15,31,data):sp_def(2,data)
1550 get(16,16,31,31,data):sp_def(3,data)
1560 sp_set(0,,,256):sp_set(1,,,257)
1570 sp_set(2,,,258):sp_set(3,,,259):wipe()
1580 sp_hang(0,1,16,0):sp_hang(1,2,-16,16):sp_hang(2,3,16,0)
1590 endfunc
1600 func int title()
1610 int k,x,y,bl,br
1620 img_scrn(1,2,1):mouse(1):msarea(210,300,282,380)
1630 for k=0 to 7
1640   symbol(195+k, 50+k,"小倉",2,2,2,120+k,0)
1650   symbol( 60+k,130+k,"百",3,3,2,120+k,0)
1660   symbol(160+k,130+k,"人",3,3,2,120+k,0)
1670   symbol(260+k,130+k,"一",3,3,2,120+k,0)
1680   symbol(360+k,130+k,"首",3,3,2,120+k,0)
1690 next
1700 symbol(203, 57,"小倉",2,2,2,130,0)
1710 symbol( 67,137,"百",3,3,2,130,0):symbol(167,137,"人",3,3,2,130,0
)
1720 symbol(267,137,"一",3,3,2,130,0):symbol(367,137,"首",3,3,2,130,0
)
1730 symbol(160,450,"Programed By Kazu Mijyou",1,1,1,150,0)
1740 palet(254,3000):palet(255,3000)
1750 symbol(216,300,"START",1,1,2,254,0)
1760 symbol(210,356,"CONFIG",1,1,2,255,0)
1770 while 1
1780   msstat(x,y,bl,br):mspos(x,y)
1790   if y<=311 then repalet(254,254,1)
1800   if y>=356 then repalet(255,254,1)
1810   if y<=311 and bl=-1 then clearer():return(0)
1820   if y>=356 and bl=-1 then config()
1830   if br=-1 then clearer():return(-1)
1840 endwhile
1850 endfunc
1860 func clearer()
1870 wipe():img_scrn(1,1,1):mouse(2)
1880 endfunc
1890 func config()
1900 dim str moji(7)={
1910 "決まり字表示","決まり字のみ"," 上の句のみ",
1920 " 下の句まで","   普通","上の句","下の句",""}
1930 dim int cp(1,3)={0,1,2,3,0,1,4,7}
1940 int x,y,bl,br
1950 vpage(2):apage(1):msarea(180,230,324,361)
1960 palet(252,3000):palet(253,3000):palet(254,3000):palet(255,3000)
1970 for j=0 to 1
1980   symbol(148,50,moji(5+j),3,3,2,150,0)
1990   for i=0 to 3
2000     symbol(180,230+i*36,moji(cp(j,i)),1,1,2,252+i,0)
2010   next
2020   Mode(j)=5:while Mode(j)=5
2030     msstat(x,y,bl,br):mspos(x,y)
2040     if y<=253 then repalet(252,252,3)
2050     if y>=266 and y<=289 then repalet(253,252,3)
2060     if y>=302 and y<=325 then repalet(254,252,3)
2070     if y>=338 then repalet(255,252,3)
2080     if bl=-1 then{
2090       if y<=253 then Mode(j)=0
2100       if y>=266 and y<=289 then Mode(j)=1
2110       if y>=302 and y<=325 then Mode(j)=2
2120       if y>=338 then Mode(j)=3
2130     }
2140   endwhile:wipe()
2150 next
2160 vpage(1):apage(0):msarea(210,300,282,380)
2170 endfunc
2180 func repalet(no,noo,k;int)
2190 for i=0 to k
2200   if no=i+noo then palet(no,60000) else palet(i+noo,3000)
2210 next
2220 endfunc
2230 func PCM()
2240 int fp
2250 fp=fopen("CHH.pcm","r"):fread(SEIKAI,319,fp):fclose(fp)
2260 fp=fopen("BD1.pcm","r"):fread(BATU,319,fp):fclose(fp)
2270 endfunc
```

リスト3 いっしゅ.DAT

```
3あきのたのかりほのいほのとまをあらみ
7わがころもではつゆにぬれつつ
3はるすぎてなつきにけらししろたへの
4ころもほすてふあまのかぐやま
2あしびきのやまどりのをのしだりをの
3ながながしよをひとりかもねむ
2たごのうらにうちいでてみればしろたへの
2ふじのたかねにゆきはふりつつ
2おくやまにもみぢふみわけなくしかの
2こゑきくときぞあきはかなしき
2かささぎのわたせるはしにおくしもの
2しろきをみればよぞふけにける
3あまのはらふりさけみればかすがなる
2みかさのやまにいでしつきかも
3わがいほはみやこのたつみしかぞすむ
3よをうぢやまとひとはいふなり
3はなのいろはうつりにけりないたづらに
4わがみよにふるながめせしまに
2これやこのゆくもかへるもわかれては
2しるもしらぬもあふさかのせき
6わたのはらやそしまかけてこぎいでぬと
4ひとにはつげよあまのつりぶね
3あまつかぜくものかよひぢふきとぢよ
2をとめのすがたしばしとどめむ
2つくばねのみねよりおつるみなのがは
3こひぞつもりてふちとなりぬる
2みちのくのしのぶもぢずりたれゆゑに
4みだれそめにしわれならなくに
6きみがためはるののにいでてわかなつむ
7わがころもでにゆきはふりつつ
2たちわかれいなばのやまのみねにおふる
3まつとしきかばいまかへりこむ
2ちはやぶるかみよもきかずたつたがは
3からくれなゐにみづくくるとは
1すみのえのきしによるなみよるさへや
2ゆめのかよひぢひとめよくらむ
4なにはがたみじかきあしのふしのまも
3あはでこのよをすぐしてよとや
2わびぬればいまはたおなじなにはなる
7みをつくしてもあはむとぞおもふ
3いまこむといひしばかりにながつきの
2ありあけのつきをまちいでつるかな
1ふくからにあきのくさきのしをるれば
2むべやまかぜをあらしといふらむ
2つきみればちぢにものこそかなしけれ
4わがみひとつのあきにはあらねど
2このたびはぬさもとりあへずたむけやま
2もみぢのにしきかみのまにまに
3なにしおはばあうさかやまのさなかづら
4ひとにしられでくるよしもがな
2をぐらやまみねのもみぢばこころあらば
8いまひとたびのみゆきまたなむ
3みかのはらわきてながるるいづみがは
2いつみきとてかこひしかるらむ
3やまざとはふゆぞさみしさまさりける
3ひとめもくさもかれぬとおもへば
4こころあてにをらばやおらむはつしもの
2おきまどはせるしらぎくのはな
3ありあけのつれなくみえしわかれより
2あかつきばかりうきものはなし
6あさぼらけありあけのつきとみるまでに
2よしののさとにふれるしらゆき
3やまがはにかぜのかけたるしがらみは
3ながれもあへぬもみぢなりけり
2ひさかたのひかりのどけきはるのひに
2しづこころなくはなのちるらむ
2たれをかもしるひとにせむたかさごの
4まつもむかしのともならなくに
3ひとはいさこころもしらずふるさとは
3はなぞむかしのかににほいける
2なつのよはまだよひながらあけぬるを
3くものいづこにつきやどるらむ
2しらつゆにかぜのふきしくあきののは
1つらぬきとめぬたまぞちりける
3わすらるるみをばおもはずちかひてし
3ひとのいのちのをしくもあるかな
3あさぢふのをののしのはらしのぶれど
3あまりてなどかひとのこいしき
2しのぶれどいろにでにけりわがこひは
2ものやおもふとひとのとふまで
2こいすてふわがなはまだきたちにけり
3ひとしれずこそおもひそめしか
4ちぎりきなかたみにそでをしぼりつつ
1すゑのまつやまなみこさじとは
2あひみてののちのこころにくらぶれば
2むかしはものをおもはざりけり
2あふことのたえてしなくはなかなかに
3ひとをもみをもうらみざらまし
3あはれともいふべきひとはおもほえで
2みのいたづらになりぬべきかな
2ゆらのとをわたるふなびとかぢをたえ
2ゆくへもしらぬこひのみちかな
2やへむぐらしげれるやどのさびしきに
5ひとこそみえねあきはきにけり
3かぜをいたみいはうつなみのおのれのみ
2くだけてものをおもふころかな
3みかきもりゑじのたくひのよるはもえ
2ひるはきえつつものをこそおもへ
6きみがためをしからざりしいのちさへ
3ながくもがなとおもひけるかな
2かくとだにえはやいぶきのさしもぐさ
1さしもしらじなもゆるおもひを
2あけぬればくるるものとはしりながら
3なほうらましきあさぼらけかな
3なげきつつひとりぬるよのあくるまは
2いかにひさしきものとかはしる
3わすれじのゆくすゑまではかたければ
3けふをかぎりのいのちともがな
2たきのおとはたえてひさしくなりぬれど
2なこそながれてなほきこえけれ
3あらざらむこのよのほかのおもひでに
8いまひとたびのあふこともがな
1めぐりあひてみしやそれともわかぬまに
3くもがくれにしよはのつきかな
3ありまやまゐなのささはらかぜふけば
2いでそよひとをわすれやはする
2やすらはでねなましものをさよふけて
2かたぶくまでのつきをみしかな
3おほえやまいくののみちのとほければ
2まだふみもみずあまのはしだて
2いにしへのならのみやこのやへざくら
3けふここのへににほひぬるかな
2よをこめてとりのそらねははかるとも
2よにあふさかのせきはゆるさじ
3いまはただおもひたえなむとばかりを
3ひとづてならでいふよしもがな
6あさぼらけうぢのかはぎりたえたえに
2あらはれわたるせぜのあじろぎ
2うらみわびほさぬそでだにあるものを
3こひにくちなむなこそをしけれ
2もろともにあはれとおもへやまざくら
3はなよりほかにしるひともなし
3はるのよのゆめばかりなるたまくらに
2かひなくたたむなこそをしけれ
4こころにもあらでうきよにながらへば
```

▶アンケートハガキの「あなたの愛機」にRED ZONEがないのはなぜ? それにしても,
RED ZONEは私の財布までもREDにしてしまいました。 和田 智(19)長野県

```
3こひしかるべきよはのつきかな
3あらしふくみむろのやまのもみちばは
2たつたのかはのにしきなりける
1さびしさにやどをたちいでてながむれば
2いづこもおなじあきのゆふぐれ
2ゆふされぱかどたのいなばおとづれて
2あしのまろやにあきかぜぞふく
2おとにきくたかしのはまのあだなみは
2かけじやそでのぬれもこそすれ
2たかさごのをのへのさくらさきにけり
1とやまのかすみたたずもありなむ
2うかりけるひとをはつせのやまおろしよ
2はげしかれとはいのらぬものを
4ちぎりおきしさせもがつゆをいのちにて
3あはれことしのあきもいぬめり
6わたのはらこぎいでてみればひさかたの
3くもゐにまがふおきつしらなみ
1せをはやみいはにせかるるたきがはの
2われてもすゑにあはむとぞおもふ
3あはぢしまかよふちどりのなくこゑに
2いくよねざめぬすまのせきもり
3あきかぜにたなびくくものたえまより
```

```
2もれいづるつきのかげのさやけさ
3なががらむこころもしらずくろかみの
4みだれてけさはものをこそおもへ
1ほととぎすなきつるかたをながむれば
2ただありあけのつきぞのこれる
2おもひわびさてもいのちはあるものを
2うきにたへぬはなみだなりけり
5よのなかよみちこそなけれおもひいる
2やまのおくにもしかぞなくなる
3ながらへばまたこのごろやしのばれむ
2うしとみしよぞいまはこひしき
2よもすがらものおもふころはあけやらで
1ねやのひまさへつれなかりけり
3なげけとてつきやはものをおもはする
2かこちがほなるわがなみだかな
1むらさめのつゆもまだひぬまきのはに
1きりたちのぼるあきのゆふぐれ
4なにはえのあしのかりねのひとよゆゑ
7みをつくしてやこひわたるべき
2たまのをよたえなばたえねながらへば
2しのぶることのよはりもぞする
2みせばやなをじまのあまのそでだにも
```

```
1ぬれにぞぬれしいろはかはらず
2きりぎりすなくやしもよのさむしろに
4ころもかたしきひとりかもねむ
3わがそではしほひにみえぬおきのいしの
5ひとこそしらねかわくまもなし
5よのなかはつねにもがもなぎさこぐ
3あまのをぶねのつなでかなしも
2みよしののやまのあきかぜさよふけて
2ふるさとさむくころもうつなり
3おほけなくうきよのたみにおほふかな
3わがたつそまにすみぞめのふで
3はなさそふあらしのにはのゆきならで
2ふりゆくものはわがみなりけり
2こぬひとをまつほのうらのゆふなぎに
2やくやもしほのみもこがれつつ
3かぜそよぐならのをがはのゆふぐれは
2みそぎぞなつのしるしなりける
3ひともをしひともうらめしあぢきなく
3よをおもふゆゑにものおもふみは
2ももしきやふるきのきばのしのぶにも
3なほあまりあるむかしなりけり
```

リスト4　画面設定.SXB

```
 1: ▼Window Size (244,164),0,0,0,画面設定
 2: /*
 3: /* ＳＸでテレビが見られる「画面設定.sxb」
 4: /*                    by 矢吾
 5: /*
 6: /* ver0.99.01 (Revision3) 1995.3 投稿用に整理（512*384を外す）
 7: /* ver0.99     (Revision2) 1995.3 あとはSX-Basicのデバッグを待
つだけ
 8: /*      (alartの戻り値を受けられず、未宣言の関数にされる。使用O
S SX-Window ver3.0)
 9: /* ver0.1      (Revision1) 1995.3 初めの第一歩
10: /*
11: ▼3,StnBtn1 (16,16,112,48),0,0,７６８＊５１２
12: /* 768*512 （普通のモード）
13: func StnBtn1_Click()
14: int i
15: TVOff()
16: i=findtskn("sadjust.r",0)
17: sendmes(i,"CRTC $00,$16,$89,$0e,$1c,$7c,$02,$37,$05,$28,$02,$2
8,$1b,$ff,$28,$e16")
18: endfunc
19: ▼3,StnBtn2 (128,16,224,48),0,0,７６８＊５７６
20: /* 768*576 アスペクト比１：１（Goddess６８Ｋ標準）
21: func StnBtn2_Click()
22: int i
23: TVOff()
24: i=findtskn("sadjust.r",0)
25: sendmes(i,"CRTC $00,$16,$87,$08,$1d,$7d,$02,$5a,$01,$19,$02,$5
9,$1b,$ff,$19,$16")
26: endfunc
27: ▼3,StnBtn3 (16,64,112,96),0,0,５１２＊５１２
28: /* 512*512（スプライト用）
29: func StnBtn3_Click()
30: int i
31: TVOff()
32: i=findtskn("sadjust.r",0)
33: sendmes(i,"CRTC $00,$15,$5b,$09,$11,$51,$02,$37,$05,$28,$02,$2
8,$1b,$ff,$28,$15")
34: endfunc
35: ▼3,StnBtn4 (128,64,224,96),0,0,512*512 ＴＶ
36: /* 512*512 （テレビ）
37: func StnBtn4_Click()
38: int i
39: i=findtskn("sadjust.r",0)
40: sendmes(i,"CRTC $00,$05,$4b,$03,$05,$45,$01,$03,$02,$10,$02,$2
8,$1b,$ff,$28,$05")
41: sendmes(i,"SIZE 480,480")
42: TVOn()
43: endfunc
44: ▼3,StnBtn5 (16,112,112,144),0,0,６４０＊４８０
45: /* 640*480 ＶＧＡもどき
46: func StnBtn5_Click()
47: int i
48: TVOff()
49: i=findtskn("sadjust.r",0)
50: sendmes(i,"CRTC $00,$16,$89,$0f,$21,$71,$02,$04,$02,$20,$02,$0
0,$1b,$ff,$20,$16")
51: endfunc
52: ▼3,StnBtn6 (128,112,224,144),0,0,１０２４*７６８
53: /* 1024*768 アスペクト比１：１（ハイレゾ）
54: func StnBtn6_Click()
55: int i
56: /*i=alart(&h0006,"ﾏﾙﾁｽｷｬﾝﾓﾆﾀｰ用のﾓｰﾄﾞです。") /*虫が取れたら.
..
57: alart(&h0006,"ﾏﾙﾁｽｷｬﾝﾓﾆﾀｰ用のﾓｰﾄﾞです。")
58: input "Yes=1 No=2";i
59: if i=1 then {
60:         TVOff()
61:         i=findtskn("sadjust.r",0)
62:         sendmes(i,"CRTC $00,$1a,$8e,$02,$07,$87,$01,$8a,$01,$0
a,$01,$88,$1b,$ff,$0a,$1a")
63: }
64: endfunc
65:
66: func TVOn() /* スーパーインポーズ
67: set_reg(1,&h0000001f)
68: iocs(&h0c) /* TVCTRL
69: endfunc
70:
71: func TVOff() /* スーパーインポーズ解除
72: set_reg(1,&h0000001d)
73: iocs(&h0c) /* TVCTRL
74: endfunc
```

リスト5　りもこん.SXB

```
 1: ▼Window Size (108,104),0,0,0,ﾃﾚﾋﾞﾘﾓｺﾝ
 2: int sp=0
 3: ▼3,StnBtn1 (8,4,28,24),0,0, 1
 4: func StnBtn1_Click()
 5: set_reg(1,&h00000010)
 6: iocs(&h0c)
 7: endfunc
 8: ▼3,StnBtn2 (32,4,52,24),0,0, 2
 9: func StnBtn2_Click()
10: set_reg(1,&h00000011)
11: iocs(&h0c)
12: endfunc
13: ▼3,StnBtn3 (56,4,76,24),0,0, 3
14: func StnBtn3_Click()
15: set_reg(1,&h00000012)
16: iocs(&h0c)
17: endfunc
18: ▼3,StnBtn4 (80,4,100,24),0,0, 4
19: func StnBtn4_Click()
20: set_reg(1,&h00000013)
21: iocs(&h0c)
22: endfunc
23: ▼3,StnBtn5 (8,28,28,48),0,0, 5
24: func StnBtn5_Click()
25: set_reg(1,&h00000014)
26: iocs(&h0c)
27: endfunc
28: ▼3,StnBtn6 (32,28,52,48),0,0, 6
29: func StnBtn6_Click()
30: set_reg(1,&h00000015)
31: iocs(&h0c)
32: endfunc
33: ▼3,StnBtn7 (56,28,76,48),0,0, 7
34: func StnBtn7_Click()
35: set_reg(1,&h00000016)
36: iocs(&h0c)
37: endfunc
38: ▼3,StnBtn8 (80,28,100,48),0,0, 8
39: func StnBtn8_Click()
40: set_reg(1,&h00000017)
41: iocs(&h0c)
42: endfunc
43: ▼3,StnBtn9 (8,52,28,72),0,0, 9
44: func StnBtn9_Click()
45: set_reg(1,&h00000018)
46: iocs(&h0c)
47: endfunc
48: ▼3,StnBtn10 (32,52,52,72),0,0,10
49: func StnBtn10_Click()
50: set_reg(1,&h00000019)
51: iocs(&h0c)
52: endfunc
53: ▼3,StnBtn11 (56,52,76,72),0,0,11
54: func StnBtn11_Click()
55: set_reg(1,&h0000001a)
56: iocs(&h0c)
57: endfunc
58: ▼3,StnBtn12 (80,52,100,72),0,0,12
59: func StnBtn12_Click()
60: set_reg(1,&h0000001b)
61: iocs(&h0c)
62: endfunc
63: ▼3,StnBtn13 (56,76,76,96),0,0, +
64: func StnBtn13_Click()
65: set_reg(1,&h00000001)
66: iocs(&h0c)
67: endfunc
68: ▼3,StnBtn14 (80,76,100,96),0,0, -
69: func StnBtn14_Click()
70: set_reg(1,&h00000002)
71: iocs(&h0c)
72: endfunc
73: ▼3,StnBtn15 (8,76,28,96),0,0, V
74: func StnBtn15_Click()
75: set_reg(1,&h00000009)
76: iocs(&h0c)
77: endfunc
78: ▼3,StnBtn16 (32,76,52,96),0,0,SI
79: func StnBtn16_Click()
80: if sp=0 then {
81:         sp=1
82:         set_reg(1,&h0000001f)
83:         iocs(&h0c)
84: } else {
85:         sp=0
86:         set_reg(1,&h0000001d)
87:         iocs(&h0c)
88: }
89: endfunc
```

▶付録ディスクが５インチからCD-ROMになっても，1,000円は超えないようにしてください。
金山　美広(23)神奈川県

1200bpsじゃ，たまらない

Takazawa Kyoko
高沢 恭子

パソコンの環境は人によってさまざまです。パソコン通信をするにも，通信速度が1200bpsの人もいれば，28800bpsの人もいます。恭子さんのマシンは1200bpsで，なにかと不便なようですが……。

会えたらいいけど

学生時代の親友「悦っちゃん」には，友人たちのあいだで，待ち合わせの約束をしてもどのくらい待たされるかわからないという定説があった。

ひどいときには3時間，あるいは半日は覚悟したほうがいいという人もいた。だんだんつくり話になって，翌日まで待ったという人も出てきた。

悦っちゃんは，本分である絵の勉強よりも，学外のことに興味の対象が多かったようで，ふつうの学生とはずいぶん質のちがう行動をしていた。

とくに演劇やパントマイムに対する関心が強く，いくつかのスタジオに出入りしてレッスンにはげんでいるようだった。授業や絵画の制作は，悦っちゃんの興味の中心からハズレていたらしい。

彼女とひんぱんに待ち合わせをしていたのは学生時代のことだから，グループや集団でのばあいが多かった。たしかに，悦っちゃんはいつも大幅に遅れてきたし，ときにはこないこともあったと思う。けれど多人数での待合せでは，誰かが個人的に困るということはなかった。

問題は，悦っちゃんとふたりだけで待ち合わせをしたばあいだ。彼女の行動のクセをよく心得ている私でも，1時間もしないうちに不安になって，そのあとの対処に困惑したものだ。

そんなふうだから，アルバイト先も無断欠勤をして，あっさりクビになったこともあったそうだ。これは，彼女が事実として淡々と私に伝えたことだ。

悦っちゃんはもともと無口で，遅れてきても言い訳はしたことがなかった。開きなおるわけでもなく，ごくあたりまえの表情をしているので，こちらも，これはなんでもないことなんだと思ってしまうような，ふしぎな空気をつくる人だった。

ふつうの感覚からハミだしているとしか思えない悦っちゃんなのに，彼女を見ていると，逆に，ひとりの人(個性)と会う目的のために，分刻みの時間を限定することのほうがまちがっているのだろうかと思わせるものがあった。

悦っちゃんのお父さんは刑事さんで，誕生日は6月10日「時の記念日」であることを知っていたのは，私くらいだ。

どうにもならないクセ

大胆な遅刻をしていたのは学生のころの話。いまの悦っちゃんは，愛知県で「火の鳥」という市民劇団を主宰し，ご主人とともに意欲的な活動をしている。

定刻から時間単位で遅れてしまうと，もう遅刻の領域ではないのかもしれない。

悦っちゃんほどのスケールの人には，その後，ひとりしか出会っていない。

ふつうよくある遅刻は，悦っちゃんにくらべたらささやかなものだ。数分から，せいぜい，30分以内のものと思う。

どうしても，かならず15分くらい遅刻するという人がいる。それなら，すべてを15分早くすればよいと思われるのに，けっしてできない。じつは私もそういう時代があった。

なぜかと自分に問う。

申しあわせを破ることに平気な習慣ができている。つまり，自分の意識が長いことつみあげてつくった，クセである。

それは，相手にどんな迷惑をあたえているかについて，きちんと考えたことがないのも原因になっている。とくに，業務や会議の開始時刻などに遅れたばあい，迷惑をかけている相手はどこにもいないような錯覚をしている。

じつは，みんなが呼吸をそろえようとしているスタートのタイミングを，自分が崩しているのに気づいていない。遅れた時間だけ，みんなが「水もれ」の感じをあじわっていることを，当人だけがわからない。

そういう人は，誰かと待ち合わせてもやはり遅れる。約束の時刻以前に到着することは，はじめからアタマにない。むしろ，定刻から何分くらいなら経過してもかまわないだろうと予定している。

そんな人でも，定刻の厳守に努力するばあいがある。たとえば指定の列車に乗るときや，先着の利点がある公演のチケットをもとめるときなどだ。

つまり，その目標に対する意欲が強いほど，遅刻の可能性はすくなくなる。いかに望んでいるか，いかに力を入れているかが行動をきめ，遅れるとなにが失われるかも要因になる。

じつはほとんどの遅刻で失われているのは，まず信用なのだが，信用が原因で事件がおこらないかぎり，その人の遅刻はきっとなおらない。

名人は待たせる

約束にもいろいろあるので，遅れることがすべて失敗になるとはかぎらない。恋人との待ち合わせ，パーティやクラス会，授業参観，それぞれにタイミングのくふうがありそうだ。

どんなぐあいに約束したかによって，登場するテクニックもあるだろう。それによる効果も損失もきっとある。

DOS/Vマシンのマウスを半年待たされた話を狛江のアニキに聞かせた。

マシンは届いたが，マウスだけすこし待ってほしいといわれて，あまり理由も知らされないまま，7月から12月まで代用品を使っていた話だ。

その結果，なめらかで小さめ，手のひらによくおさまる，なかなかの使いごこちのマウスが届いた。DOS/V専門ショップのＹ氏が，機器の吟味にこだわったあげくのことだった。

「イタリアだったか，ギターの名器づくりの職人にある人が注文したところ，17年かかって届いたという話があるよ」とアニキがいった。「ほんとうなら30年はかかるのだが，あなたのは，ちょっと早めにしたんだ」といったそうだ。

レオナルド・ダ・ヴィンチに絵をたのむと，何年たっても絵の具づくりをしているばかりで，制作をはじめなかったという話もある。

はじめから，もとめているものに価値を感じて待つばあいと，待っているあいだに価値を感じはじめるばあいがある。ギターづくりの名人は，前者に入るとしても，マウスのほうはどうだったのだろう。待つあいだにだんだん要求が強くなって，マウスひとつの価値が強調されたようにも思うのだけれど。

つい数日前も，１カ月がかりで手に入れたものがある。

FBI-NETのイベントボードに，ジャーナリストのPATA氏のコメントがあった。「;P;80;WINに，先日のパーティで撮った写真３枚をアップしました。BMPファイルにしてありますんで，Windowsの壁紙にでもしてください。あ，500Kあるんですよ，256です」。

あるオフ会でPATA氏が撮影した写真を，PDSのボードにBMPファイルでアップしたという知らせだった。さっそくダウンロードした人たちから，よくとれている，スゴイという感想がつぎつぎアップされはじめた。

オフ会はシスオペの中村隊長とPATA氏，ちゃがま氏の企画でおこなわれたパーティで，FBIやNIFTY-Serveのかたたちが多数参加した。

そのときの写真がパソコンの画面で再現されてみんなが歓声をあげているのに，私だけトライできないのは残念だ。

いそがば郵便

通信の速度が威力を見せるのは，やはり大量のデータを送受するときだ。

いまだに1200bpsの東京の家でダウンロードすると，500Kバイトを記録するのに80分かかる。都内でふつうに通信をしているときはなんの不便も感じないが，バイナリファイルのダウンロードとなるとこういうことになる。

それでも，80分かかるとなると，なおのこと，その写真ファイルがほしくなる。

PDSボードの〈80〉で，80級のCPUを指定，[W]でMS-WINDOWSを選択，[R]でダウンロードの要求をした。

＊＊バイナリファイル 541568バイト＊＊と内容量の表示があり，ダウンロードしますかとの確認がある。「Y」の入力でダウンロードの態勢となる。

プロトコルでXMODEMを選び，ファイル名をつけてダウンロードを開始。

ところが延々とすすめるうち，なんと半分ほどいったところでエラーがおこった。不覚にも座礁である。これもパソコンの習性とナットクしても，また80分にいどむ闘志はうすれがちだ。

あきらめかけてアクセスすると，イベントボードで話題はつづいている。

「じつは画像処理はプロにやってもらったんです。ハードもふくめて環境がスゴイもんで。フラッシュ撮影はアンダー気味でネガの状態はよくなかったんですが，あそこまで，見られるようにしてもらいました」などとPATA氏の説明もある。

カメラの話題も出はじめた。ちょうど先日，弟が新宿の家から自分の使っていたミノルタの１眼レフを持ってきて，「数がふえちゃったんで，１台使ってみない？」と置いていったところだ。

ついにPATA氏に直接メールをして，コピーしたディスクを送っていただくことにした。通信手段の敗北である。

そして，はじまりからひと月，やっとファイルはWindowsのSUPERKID上で開かれた。待っていた時間の効果もあったかもしれないが，やっぱり歓声をあげたくなるような仕上がりだった。PATAさん，ほんとにありがとう。

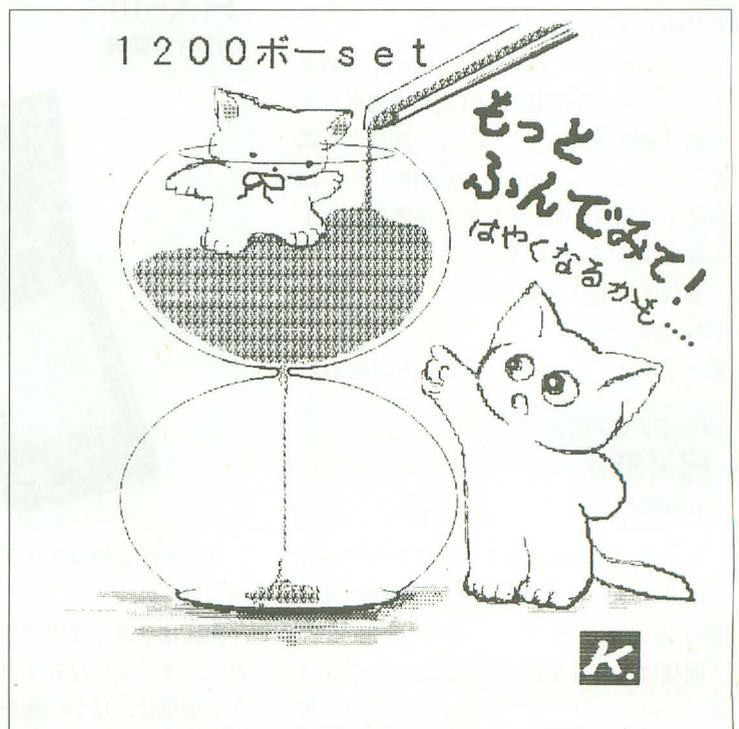
三重の住いの近くにある写真屋さんに，たのんでおいた写真を取りにいった。

「あのう，ずいぶん辛抱づよいですね」と写真屋さんのご主人がいった。「撮りはじめから終わりまでに，長いことかかってますよねぇ」というのだ。

「ええ，あんまりひんぱんには写さないものですから」

帰宅して写真を見ておどろいた。１本のフィルムで，最も古い日付が「87 1 1」，最新の日付は「95 3 12」となっている。

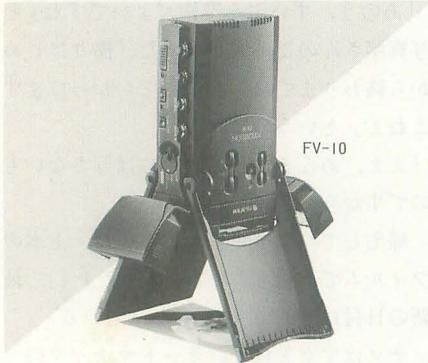
ナゾはすぐとけた。オートマチックのカメラの電池を交換したときリセットされて，87年の日付にもどったのだ。

写真屋さんがフシギな顔をしていたのもむりはないが，いくらなんでも，８年かけて１本のフィルムを消費するガマン強さは持てそうにない。

illustration：Kyoko Takazawa

PENGUIN INFORMATION CORNER

ペ・ン・ギ・ン・情・報・コ・ー・ナ・ー

NEW PRODUCTS

フォトビジョン FV-10 富士写真フイルム

FV-10

富士写真フイルムはフォトビジョン「FV-10」を発売した。

同機はネガフィルム/スライド/プリント/印刷物などをテレビに映したり，パソコンに画像入力したりできる。また，カメラモードでは，人物や風景など大きな被写体や動画も取り込むことが可能。

レンズは1/4インチの41万画素CCDフジノンレンズを搭載し，CCDからのデジタル信号をDSP回路で処理して，露出や色調整，ピント合わせなどを自動制御する。出力端子はS映像端子とビデオ映像端子を1系統ずつ装備している。

価格は79,800円(税別)。

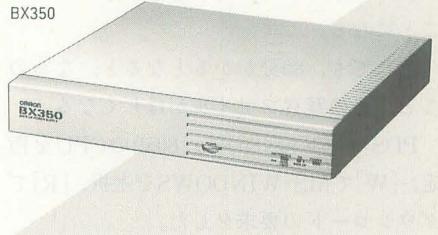
〈問い合わせ先〉

富士写真フイルム㈱　☎03(3406)2111

バックアップ電源 BX350 オムロン

BX350

オムロンは停電時などの電源トラブル時に電源を供給するバックアップ電源「BX350」を発売した。

同機は出力容量が350VA(210W)で5分間以上のバックアップが行える。電源の切り替え時間は10msで，バッテリーの充電時間は8時間。出力はバックアップ出力が2系統，スルー出力が1系統。また，バックアップ電源使用時やバッテリーロー，オーバーロード時にはLED表示とブザーで警報を発する。

大きさは285mm(幅)×287mm(奥行)×45mm(高さ)で，重さが4.5kgと従来機「BX3」に比べて体積比30%減，重量比10%減を実現した。

価格は24,800円(税別)。

〈問い合わせ先〉

オムロン㈱　☎045(411)7223，06(282)2672

電子手帳 RX-10 カシオ計算機

RX-10

カシオ計算機は電子手帳「RX-10」を発売した。

同機は手書き文字認識を内蔵した電子辞書で，すべての操作を付属のペンで行う。主な機能は，ほかの機能を使っているときでも呼び出せる即メモ機能，合計約128,500語の英和/和英/国語辞典機能，お金の出入りを管理できるマネーシート機能，住所録機能などを搭載している。ほかにも，スケジュール，世界時計，カレンダー，カードゲーム，検索機能などが用意され，各機能の使い方がわからない場合も，ヘルプ機能によるアドバイスが受けられる。

記憶容量は約110Kバイト。単4形アルカリ電池2本で約70時間の使用が可能。

大きさは蓋を閉じた状態で81.9mm(幅)×126mm(奥行)×13.5mm(高さ)，重さが149g(電池含む)。

価格は35,000円(税別)。

〈問い合わせ先〉

カシオ計算機㈱　☎03(3347)4811

辞書電卓 CW-400 セイコー電子工業

CW-400

セイコー電子工業は辞書電卓「CW-400」を発売した。

同機は英和/和英の2つの辞書機能を搭載している。英和の見出し語が約7,500語，訳語約9,500語。和英は見出し語が約7,000語で訳語が約9,000語となっている。電卓機能は，8桁1メモリの四則演算が可能なほか，通貨/単位換算機能がついており，通貨や長さ，体積，重量，温度などの換算が可能。ほかの機能は，通常の時計や世界時計，カレンダー，アラームなどがある。

大きさは蓋を閉じた状態で108mm(幅)×67mm(奥行)×12mm(高さ)，重さが約65g

(電池含む)。

価格は5,000円(税別)。

〈問い合わせ先〉

セイコー電子工業㈱　　　✆0120(052)440

電子ブックプレーヤー
DD-55
ソニー

DD-55

ソニーは電子ブックプレーヤー“三省堂辞書10巻”「DD-55」を発売した。

同機は(株)三省堂の10巻の辞書を1枚のディスクに収めた電子ブックソフトとデータディスクマンをセットにしたものである。従来機に加え，ソフトを挿入すると自動的に電源が入ったり，ソフトの使い方を音声と図で説明する機能などが追加されている。重さも従来機に比べ約100g軽く，約480g(電池，ソフト含む)を実現し，電池寿命も約2倍の14時間の再生が可能になった。また，収録された辞書は「ニューセンチュリー英和辞典」「新クラウン和英辞典　第5版」「三省堂現代国語辞典」「必携故事ことわざ辞典」など。

価格は35,000円(税別)。

〈問い合わせ先〉

ソニー㈱　　☎03(5448)3311，06(251)5111

BOOK

X68000 ゲーム・プログラミング
技術評論社

技術評論社はX68000用のプログラミングテクニックブック「X68000ゲーム・プログラミング」を発売した。

本書は3章に分かれ，1，2章では対話形式でC言語によるシューティングゲームを作成しながら，さまざまなテクニックを紹介していく。3章では応用編と題してスプ

X68000
ゲーム・プログラミング

ライトの増殖に挑戦したり，例外処理や常駐プログラムについてアセンブラも駆使して解説している。執筆は京大マイコンクラブによるもの。付録ディスク2枚には本書で作成したゲームを含む4本のゲームプログラムなどが入っている。

価格は4,500円(税込)。

〈問い合わせ先〉

㈱技術評論社　　　☎03(3225)2300

INFORMATION

The New Digital Age ～マルチメディアが開くあすへの扉
機械産業記念事業財団

機械産業記念事業財団は，1995年4月28日～12月25日まで東京都港区北青山のTEPIA(機械産業記念館)にて，TEPIA第8回展示「The New Digital Age～マルチメディアが開くあすへの扉」を開催する。

展示内容は，

1)　Internet World

ホームページの情報をメインにインターネットを紹介

2)　マルチメディアを支える技術

画像，音声，通信，コンピュータの各分野における最先端技術を紹介

3)　社会に広がるマルチメディア

VOD(ビデオ・オン・デマンド)，医療情報データベース，マルチメディア語学教育システム，見えるラジオなど，公共/行政/保健/エデュテイメントに関わる機器やソフトの体験コーナー

4)　インターネット体験コーナー

インターネットに自らアクセスできる体験コーナー

5)　産業に広がるマルチメディア

PDA(携帯情報端末)，画像電送対応デジタルスチルカメラ，景観シミュレーションなど，産業分野のマルチメディアでの応用例を紹介

6)　海外展示

海外におけるマルチメディアの最新情報を機器とともに紹介

開館時間は平日が10:00～18:00，土曜日・祝日が10:00～17:00，毎週日曜日は休館。入場は無料。

〈問い合わせ先〉

㈶機械産業記念事業財団　☎03(5474)6111

Computer Visualization Contest
日経サイエンス社

日経サイエンス社は各分野の研究者，学生を対象に，教育，観測，計測，集計，統計など科学技術計算の結果をコンピュータで可視化した作品を募集して審査する「Computer Visualization Contest」を開催する。

主な応募要項は以下のとおり。

実施期間：応募申込締切日　6月2日
　　　　　作品提出締切日　8月21日

募集作品対象：

(1)実験計測結果の可視化：宇宙，地球物理，気象などの計測結果や風洞実験などの実験結果ほか

(2)解析結果の可視化：有限要素法解析，流体解析，電磁場解析などの科学技術計算結果，化学分野における分子形状ほか

(3)画像処理/ボリュームレンダリング：医療画像，画像認識などの画像処理やボリュームレンダリングほか

(4)データ解析，プレゼンテーションほか：景観設計，CADデータ表示/加工，地図情報システムほか

(5)その他ビジュアライゼーションに関する作品

注意：創作CG作品は，本コンテストの対象から除く

審査：有識者6人で編成される「Computer Visualization Contest審査委員会」による厳正な審査

賞：最優秀賞～AVS大賞(副賞20万円)ほか

なお，コンテストに関する問い合わせ，応募要項の請求は下記の連絡先まで。

〈問い合わせ先〉

Computer Visualization Contest事務局

☎03(3233)3475

創刊13周年記念

愛読者特大モニタ

愛読者モニタ

A X68000 CompactXVI

1名

注）今回のモニタは本体のみでディスプレイは含まれません

▲シャープ ☎03(3260)1161

B SX-WINDOW 開発キット Workroom SX-68K

1名

X68000用

3.5″/5″2HD版

▲シャープ ☎03(3260)1161

C SX-WINDOW デスクアクセサリ集

2名

X68000用

3.5″/5″2HD版

▲シャープ ☎03(3260)1161

D XDTP SX-68K

2名

X68000用

3.5″/5″2HD版

▲シャープ ☎03(3260)1161

モニタの応募方法

希望するモニタ記号をとじ込みのアンケートハガキの左下のスペースまたは官製ハガキに記入してお申し込みください。応募の際に使用環境を明記する必要はありませんが，当選された方にはモニタとして使用ののちレポートを提出していただきます。締め切りは1995年6月18日の到着分までとし，当選者の発表は1995年8月号で行います。また，雑誌公正競争規約の定めにより，当選された方はこの号のほかの懸賞に当選できない場合がありますので，ご了承ください。

&プレゼント

E

EGWord
SX-68K

X68000用　2名

3.5″/5″2HD版

▲シャープ　☎03(3260)1161

F

SOUND
SX-68K

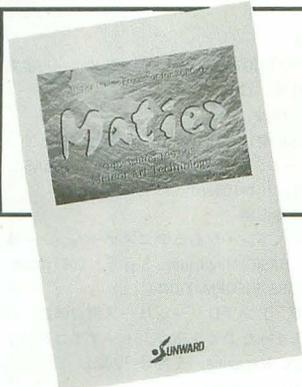
X68000用　1名

3.5″/5″2HD版

▲シャープ　☎03(3260)1161

G

MATIER ver.2.1

1名

X68000用

5″2HD版

▲サンワード　☎044(855)4335

愛読者プレゼント

1

倉庫番リベンジ
SX-68K ユーザー逆襲編

2名

X68000用

5″2HD版

▲シャープ　☎03(3260)1161

2

ディグダグ/ディグダグII

3名

X68000用

5″2HD版　5,300円（税別）

▲電波新聞社　☎03(3445)6111

プレゼントの応募方法

とじ込みのアンケートハガキの該当項目をすべてご記入のうえ，希望するプレゼント番号をハガキ右下のスペースにひとつ記入してお申し込みください。締め切りは1995年6月18日の到着分までとします。当選者の発表は1995年8月号で行います。また，雑誌公正競争規約の定めにより，当選された方はこの号のほかの懸賞に当選できない場合がありますので，ご了承ください。

4月号モニタ当選者

A XL/Image（広島県）馬場　徹太郎　B XDTP SX-68K（茨城県）伊藤　盛人

4月号プレゼント当選者

1 パックランド（神奈川県）安武　隆行（千葉県）澤田　恭幸（東京都）佐井川　泰治 2 STANDARD MIDI FILE CLASSIC SERIES　A（埼玉県）鈴木　道明　B（東京都）高橋　明　C（北海道）小谷　雅樹　D（埼玉県）加藤　洋介　E（石川県）前川　申　F（栃木県）寺田　亮治
（敬称略）

以上の方々が当選しました。商品は順次発送いたしますが，入荷状況などにより遅れる場合もあります。

FILES Oh!X

このインデックスは，タイトル，注記——著者名，誌名，月号，ページで構成されています。滅入るのは天気だけにして頑張りましょう。

一般

▶NEWS
エプソンのDOS/V機発売，インターネットを使った就職活動，アマチュアCGAコンテストの結果などのニュース。——編集部，ASAHIパソコン，4・15号，8-11pp.

▶特集1　PIMソフトで情報管理の達人になろう
パソコンで使えるPIM(個人情報管理)ソフトを紹介する。——編集部，ASAHIパソコン，4・15号，16-29pp.

▶特集2　注目サブノート3機種を徹底比較！
PC-9821Lt, DynaBook SS450, FMV-BIBLOの3機種の携帯性，画面の見やすさ，操作性などを徹底比較する。——編集部，ASAHIパソコン，4・15号，30-37pp.

▶98ユーザーのためのマッキントッシュ教室　11
今回はMacintoshのOSとWindowsを比較する。——荻窪圭，ASAHIパソコン，4・15号，108-111pp.

▶THE NEWS FILE
「GAME EXPO'95」「国際航空宇宙展」のレポートや「日本ソフトウェア大賞'94」の受賞作品発表のニュースなど。——編集部，LOGIN，8号，48-53pp.

▶ぼ，ぼ，ぼくらはパソコン鑑定団
パソコン本体や周辺機器，ソフトウェアを鑑定する知識を教える。——編集部，LOGIN，8号，123-135pp.

▶これからはソウルだ!!
「ニューメディアワールド'95」のレポートなど韓国のパソコン事情を紹介する。——編集部，LOGIN，8号，174-177pp.

▶くねくね科学探検　第18回
最後の宮大工である西岡常一氏の著書を参考に木の特性を考える。——鹿野司，LOGIN，8号，192-195pp.

▶特集　パソコンゲームを120%楽しむ方法
過去に発売されたゲームのいろいろな楽しみ方を紹介。——編集部，コンプティーク，5月号，19-29pp.

▶巷のメディアミックス
今回は，最近のゲームにかかせない声優を，出演した作品とともに紹介する。——編集部，コンプティーク，5月号，120-123pp.

▶こだわりゲーム年代記
パソコンゲームの歴史を1社の作品の流れから考える。今回はガイナックス。——与志田拓実，コンプティーク，5月号，126-127pp.

▶1/4095の選択
コンシューマ機とパソコンをひっくるめ，今後のハード，ソフトの動向を予想して，ベストチョイスを考える。——編集部，電撃王，5月号，36-51pp.

▶Arcade Game Graffiti　第15回
1982年に登場したアーケードゲームを振り返る。「ゼビウス」「バトルクロス」などを紹介。——編集部，マイコンBASIC Magazine，5月号，142-145pp.

▶'95 AOUショーレポート
各社の最新アーケードゲームを紹介する。——開田清隆，マイコンBASIC Magazine，5月号，156-159pp.

▶NEWS & VIEWS
Macintosh互換機を発売するパワー・コンピューティング社の取材レポートと他社の互換機の紹介。——編集部，ASAHIパソコン，5・1号，14-19pp.

▶98ユーザーのためのマッキントッシュ教室　12
「狭い家と画面は賢く使おう」と題して，ウィンドウ上でのいろんな操作方法を説明する。——荻窪圭，ASAHIパソコン，5・1号，102-105pp.

▶特集1　パソコンパワーアップ大作戦
CPUのアップグレードについて機種別に解説。——山口学ほか，ASAHIパソコン，5・1号，22-33pp.

▶特集2　携帯情報端末PDA新時代がやってくる
アメリカと日本のPDA事情を紹介する。登場する機器は「Newton」「MagicLink」「ザウルス」など。——山本直樹・須藤慎一，ASAHIパソコン，5・1号，107-115pp.

▶TEST RESULTS
キヤノンのカラープリンタ「BJC-35v」などの試用レポート。——編集部，ASAHIパソコン，5・1号，118-127pp.

▶ビデオ信号の「謎」
ディスプレイのアナログとデジタル信号の違いについて解説する。——赤童子，I/O，5月号，40-43pp.

▶MultiMedia Watching　17
デジタルスチルカメラ，次世代ゲーム機，デジタル放送などマルチメディアに関する話題についてコメントする。——奥野雅之，I/O，5月号，71-73pp.

▶特集　低価格レーザー・プリンタ
各社の10万円前後のレーザープリンタを機種別に紹介。——編集部，I/O，5月号，91-101pp.

▶ハードディスクの基礎知識
ハードディスクとそのインタフェイスについて基本的な部分を解説する。各社ハードディスクの一覧表つき。——英斗恋，I/O，5月号，102-110pp.

▶New Products
リコーのデジタルスチルカメラ「DC-1」，プレクスターの6倍速CD-ROMドライブ「PX-65CS」など周辺機器の情報。——編集部，I/O，5月号，156-159pp.

▶特集I　これで分かった！　パソコン新語大辞典
CPU，外部記憶装置，拡張ボード，通信などの項目についてパソコンに関連する語句を解説する。——編集部，ASCII，5月号，265-288pp.

▶特集II　最新ギガ・ストレージ接続ガイド
大容量ハードディスクの接続法をレポートする。各社ドライブの製品紹介やベンチマークテストのレポートもあり。——編集部，ASCII，5月号，299-323pp.

▶Wozの魔法使い　第3回
今回はApple IIに搭載された2つのROMに入っているプログラムを紹介する。——柴田文彦，ASCII，5月号，371-373pp.

▶魅惑のニューテクノロジー
ビデオアクセラレーション技術においてソフトウェアのキーとなるDCIについて解説していく。——編集部，ASCII，5月号，378-383pp.

▶INTERNET膝栗毛　ROUTE 4
インターネットに関する話題。新wwwブラウザの紹介など。——編集部，ASCII，5月号，388-392pp.

▶稀代もののけ考
実用小物からホビーグッズまで，巷で見つけた小物を紹介する。今回は木製マウスや御影石のマウスパッドなど。——編集部，ASCII，5月号，458-459pp.

▶日本ソフトウェア大賞'94
1994年度に発売された優秀なソフトを決める。大賞はWindows用ソフトで「デイジーアート」。——編集部，LOGIN，9・10合併号，6-15pp.

▶特集　キミのパソコンライフは間違っちょる！
パソコンライフに関する質問の集計結果を大公開する。——編集部，LOGIN，9・10合併号，163-177pp.

▶ログイン不思議発見伝　通信カラオケの謎
通信カラオケに関する疑問に答える。——編集部，LOGIN，9・10合併号，216-219pp.

▶架想楽園へ行こう Ver.2.04
現在の家庭用VR機を取り上げ，その未来について考える。——中田宏之，LOGIN，9・10合併号，220-223pp.

▶くねくね科学探検隊　第19回
前回に続いて木造建築の技術について解説する。——鹿野司，LOGIN，9・10合併号，232-235pp.

X1/turbo/Z

X1シリーズ

▶Love Triangle
列を消していくパズルゲーム。——ステキナ8BIT愛好会，マイコンBASIC Magazine，5月号，103p.

X68000

▶SUPER SOFT INDEX
機種別の新作予定表。X68000用は「地球防衛 Miracle Force」など。——編集部，コンプティーク，5月号，101-102pp.

▶電撃新作予定表
X68000用は「EXCITINGみるく 第1話」など。——編集部，電撃王，5月号，170p.

▶MAGICAL PARASOL 2
おじゃまキャラを避けながらパラソルを撃ち落とすシューティングゲーム。——宮地周仁，マイコンBASIC Magazine，5月号，104-105pp.

▶マリ夫の冒険 tennis編
1人プレイと2人プレイがあるテニスゲーム。——高橋秀之，マイコンBASIC Magazine，5月号，106-108pp.

▶SOFTWARE HOUSE INFORMATION
「なにわ通信」で，アンソロジーシリーズ第13弾に「バラデューク」が決まったことを紹介。——マイコンソフト，マイコンBASIC Magazine，5月号，202p.

▶SUPER SOFT Hot Information
X68000用は「R.C.ロボット集＋αvol.5」などが予定表のみで紹介されている。——編集部，マイコンBASIC Magazine，5月号，とじ込み付録13p.

▶AV STRASSE
「フォント＆ロゴデザインツール書家万流 SX-68K」と「学研統合電子辞書 for SX-Window」のレビュー。——編集部，ASCII，5月号，401-404pp.

▶ONLINE SOFTWARE INDEX
大手ネットにアップロードされたプログラムを紹介する。X68000用はMOをIBMフォーマットする「MOfmt.r」，拡張フロッピーディスクドライバ「9SCSET」など。——編集部，ASCII，5月号，500-501pp.

▶SX-WINDOWプログラミング　第20回
今回は前回に作成したファイルカッターに付加するファイル選択用のダイアログを作成する。——吉野智興，C MAGAZINE，5月号，140-145pp.

ポケコン

PC-E500

▶BOY MEETS GIRL
反対方向に動く少年と少女を出会わせる，パズルゲーム。——はたらいたはらたいら，マイコンBASIC Magazine，5月号，109p.

参考文献

I/O　工学社
ASAHIパソコン　朝日新聞社
ASCII　アスキー
コンプティーク　角川書店
C MAGAZINE　ソフトバンク
電撃王　主婦の友社
マイコンBASIC Magazine　電波新聞社
LOGIN　アスキー

●●●●●●●●● QUESTION and ANSWER ●●●●●●●●●

Oh!X 質問箱

付録ディスクにあったPushBON!.SXBがどうしても うまく起動できません。バグ報告があるかと待っていたのですが，なにもないようですし，私の環境がおかしいのでしょうか。それとも，なにか特殊な操作が必要なのでしょうか。　愛知県　田村竜一

これはSX-BASICが必要なデータを見つけてきていないことが原因と思われます。プログラムがかなり深いディレクトリの中にありませんか？　パス名が長すぎて，プログラムで設定されている文字列長を超えてしまっている可能性が大です。

たとえば，そのプログラムが“A:¥Oh!X付録ディスク¥XLImageお試し版＋α¥DISK2¥SXピコピコ¥PushBon!.SXB”のようなパスに保存されていた場合なら，

str mapfile＝"Push_1.map"
str pcmpath＝"",ppath

の変数宣言の部分を，

str mapfile[90]＝"Push_1.map"
str pcmpath[90]＝"",ppath

のように変更してみてください。

（中野修一）

最近安くなってきた1GバイトクラスのHDDはX68000につながりますか？　松戸市　小野和男

最近のSCSIのHDDユニットの価格低下には目を疑うようなものがあります。去年の暮れ，540MバイトクラスのHDDが5万円を切ったか！と思ってたら，いま（4/17現在）秋葉原のDOS/Vショップなどに行けば，それが19,800円で手に入ります。1Gバイトはやっと5万円切ったところで，730Mバイトが28,800円(しかもこれは爆速HDD）ってところ。ただし，ベアドライブ（ドライブ裸のまま）ですからね。したがって，ケーブルなどを自作しないとつながりません。

古いSCSIのHDDユニットなどを持っている場合や，SUPERやXVIユーザーなどの場合，80MバイトクラスのHDDなら，外付けを追加するよりも，HDDを入れ換えたほうがいいかもしれませんね。

ここで安価なGバイトクラスのHDDに目を向けます。これらのHDDは一応，X68000シリーズでも使えることは使えるのですが，残念ながら多少制限があります。X68000の制限ではなくてHuman68k側の問題です。

たとえば有名なHDD，QuantamのEMPIRE1080S（1080Mバイト）を，X68000につなげたい場合。「男は黙って1パーティション」なあなたなら，問題なく使えます。ばっちりです。

しかし，「純真な乙女は計画性がなくっちゃ」なあなたは，パーティションの切り方を多少考えなくてはいけません。

この1080S，Human68kでフォーマットすると1029Mバイトで認識します。このHDDを1024Mバイト＋5Mバイトと分けると，このHDD，フォーマット＆領域確保はできても，起動できなくなってしまいます。それどころか，FORMAT.X以外では認識できなくなってしまうのです。

簡単な話，2つ目以降のパーティションの始まりが1024Mバイト以上の場合，この障害が起きるのです。1023Mバイト以下は，どのように分けても構いません。1024Mバイト以上に，Human68kが入ってなければこの問題は起きませんから，これを利用して，巷の大容量HDDユーザーは後ろにNetBSDなどを置いておくそうです。

1029MバイトHDDならともかく，2Gバイト，4Gバイトクラスになると多少困るかもしれません。

なお，余談ですが，Human68kでは1パーティションが1024Mバイトを超えると，1クラスタが32Kバイトにもなります。512M～1023Mバイトまでは16Kバイト。クラスタサイズが大きくなると，HDDの転送速度が上がる半面，たった1バイトのファイルでも，必ず32Kバイトの容量を食うことになります。32Kバイト＋1バイトの場合などでも，64Kバイト無条件に取られます。

これが4Gバイトにもなると1クラスタは128Kバイトになりますから，恐ろしい話ですよね。きちんとこれがHuman68kで認識できるのか，試す環境がないのが残念ですけど。

ただ，クラスタサイズが上がれば，ハードディスクが吹っ飛んだときになんらかの方法でファイル復活を試みる場合，必ずクラスタ単位で復活できるんですよね。32Kバイトなら1個クラスタを修復すると，たいていのテキストは復活できたりするという利点もあります。

最近のHDDは小細工なしでも速いですし，どっちを取るかは人それぞれでしょう。1029Mバイト確保できるHDDの場合，1パーティション派の人なら1023Mバイト確保し，6Mバイト無駄にしたほうが，実質たくさんのファイルを入れることができるかもしれません。

え？　男は小さなことは気にしない？まぁ，ごもっともで。　（瀧　康史）

質問にお答えします

日ごろ疑問に思っていること，どんなことでも結構です。どんどんお便りください。難問，奇問，編集室が総力を挙げてお答えいたします。ただし，お寄せいただいているものの中には，マニュアルを読めばすぐに解答が得られるようなものも多々あります。最低限，マニュアルは熟読しておきましょう。質問はなるべく具体的に機種名，システム構成，必要なら図も入れてこと細かに書いてください。また，返信用切手同封の質問をよく受けますが，原則として，質問には本誌上でお答えすることになっていますのでご了承ください。なお，質問の内容について，直接問い合わせることもありますので電話番号も明記してください。

宛先：〒103　東京都中央区日本橋浜町3-42-3
ソフトバンク株式会社出版部
Oh!X編集部「Oh!X質問箱」係

FROM READERS TO THE EDITOR

じめじめした天気が続く，嫌な季節になりました。雨ばかりでは外出するのがためらわれるし，ついついテレビばかりを見てしまう人も多いのでは？　でも，せっかく家でゆっくりできるのだから，ポコポコプログラミングに励みましょう。

◆4月号の特集を読んでから，部屋の暗黒領域においやられていた古いゲームたちを遊んでみたが，どれもまだまだ面白い。特に「グラディウスII」はいま見てもやっぱりすごい。こうしてみるとX68000のゲームには，良作が多いことに改めて気づかされる。今後もよいゲームがどんどん出てきますように……。

松尾　泰之(23)大阪府

◆4月号の特集「Let's Play Wonderful GAME」には笑わせてもらいました。L○GiNでは，発売されるソフトの数のせいなのか，誌面から姿を消してしまったX68000ですが，今後もいままでのようなソフトが次々と発売されれば……と思っています。X68000にはまだまだがんばってもらいたいものです。　星　武史(20)山形県

◆4月号の特集を読んでいて懐かしさを感じた。「グラディウス」「スペースハリアー」「R-TYPE」「ドラゴンスピリット」に燃えた頃もあったんだなあ，と思い出しながらゲームを起動しようと思っています。　井上　一夫(37)東京都

特に思い入れのあるゲームはいつ遊んでも面白いですよね。

◆いやあ，まさか「ダッシュ野郎」が取り上げられるとは思いませんでした。私は東亜プランの大ファンなので嬉しいです。しかし，瀧さんの「B級シューターへの道」で，東亜シューティングが出てこないのは気になります。

山中　祐司(19)栃木県

ちょっとだけ好みのゲームのベクトルが瀧氏と違っていたのですね。

◆なんだかんだいっても，4月号の特集にあったゲームレビューは楽しめました。でも，発売当時にあのノリで紹介していたら，Oh!Xのゲーム批評もほかの雑誌と比べて，ひと味違っていたでしょうにねえ。　河野　太郎(22)東京都

いくらGAME OF THE YEARだからといって，特集はちょっとだけ飛ばしすぎたかな，と思っていたのですが，気に入ってもらってなによりです。

◆いやあ，ザインには笑わせてもらった。記事を読んだあと，思わず中古屋に探しにいってしまった。それにしても，記事までザインネスを漂わせている。なにしろ誤字脱字どころか文章がないんだもの。　平間　大輔(17)北海道

痛っ，ハガキを読んだ途端なんか心臓に衝撃が……。

◆4月号のザインソフト関係の記事で思い出したのですが，「未来」の最終ステージの中ボス（前半の）「サリオン」がとうとう倒せませんでした。彼を倒すことはできるのでしょうか（どんなにショットを当ててもヒット判定が出ない)。もしもこのゲームをクリアした方がいらっしゃいましたら，ぜひ方法を教えてください。

黒部　浩孝(22)三重県

このほかにも回答を求めるハガキが1通。求む！「未来」完全攻略ガイド。

◆なぜいま！　しかもザイン！　Oh!Xのネタ不足を心配しつつ大爆笑させていただきました。いまでは笑って許せるク○ゲーって感じですか。

八亀　桂一(20)神奈川県

そう，いまとなってはいい思い出なのですよ。

◆いままでイロモノ（お色気ものではない）といえば，CDやLDなどでしか知らなかった私ですが，「ザインで勝負」を読んで新たなイロモノの世界を知りました。あ，ザインソフトはイロモノではありませんでしたね。これは失礼。という私の初めてのRPG（ポッ）は，「トリトーン」だったりします。　森口　賢(22)福岡県

「トリトーン」はそこそこよかったのですが，なぜX68000に参入した途端……どこで道を踏み外してしまったのでしょうね。

◆4月号の記事につられてザインソフトを2本買ってしまった。その結果，Oh!Xの記事に誤りのないことを改めて知った。どど～ん！

松尾　繁(21)福岡県

身をもって真実を確かめる松尾さんの心意気に，ちょっとだけ敬服しちゃいますね。

◆「ザインで勝負！」の「アルフェイム」の記事がずっぽりと抜けているのは，はたして事故なのか，故意なのか？　場所が場所だけに判断に苦しむ。いまのところ「故意を装った事故」説が有力。　鹿又　健(25)東京都

……(もはやなにもいえない)。

◆本当にザインはどこにいっちゃったんでしょう？　ウィ○キーはバンプレストで，チャンピ○ンはその筋のトップ争いしてますけど。そういえば，8年ほど前，友人が「俺は大学をやめてザインに入る！」といっていたが，結局大学側がやめさせてくれなくて(?)ザイン入りがパーになった，つうことがありました。いま考えるとそれでよかったんでしょうね。

久米　豊信(26)群馬県

なかなか激しい友人だなあ。

◆1994年度のOh!Xゲーム大賞は，やはり「ジオグラフシール」になりましたか！　投票してよかったです。新作の話は聞きませんが，これからもがんばってほしいですね。あとザイン関係の記事は爆笑させてもらいました。X68000ユーザーとして，いままで知らなかったのが残念(?)なくらいです。これからもX68000に個性的なゲームが出るといいのですが。

武藤　一文(22)埼玉県

あまり個性的すぎてアクが強すぎても困っちゃいますけど。

◆「ジオグラフシール」面白そうですねえ。

竹川　貴彦(17)愛知県

◀岩瀬　貴代美　福岡県　最近，秘かに歳を重ねていくOh!X。そんななかでも，行事ごとのイラストを忘れないのは，さすがOh!Xイラストクイーン。ありがとうね。

◀溝畑　知幸　兵庫県　ゲームは色あせても，溝畑さんの描くブロウィンはいまでも生き生きしているなあ。うむ，かあいくてよろしい。

本当に，面白いですよ～。

◆おそらくX68000にとっては最後になるであろう今年のGAME OF THE YEARで，X68000に縁の深いエグザクトが選ばれたことはとても喜ばしいことだと思います。エグザクトさんには次世代「X」でもがんばってもらいたいものです。

松永　正弘(25)京都府

現在はPlayStationでがんばっているエグザクト。これからもユーザーの心をつかむ面白いゲームを作り続けてほしいですね。

◆今年のGAME OF THE YEARは，「スーパーストリートファイターII」の健闘ぶりが非常に印象に残りました。メインメモリ要4Mバイト，16MHz以上という，ユーザーを選ぶ仕様であるにもかかわらず，ここまで支持されるとは思いもよりませんでした。6月号のアンケート分析大会のメインメモリ容量にどのような変動が見られるか楽しみですね。ところで，昨年のGAME OF THE YEARを見たくなって，読み直しているとエルフの「同級生」が支持されているのが気になりました。大賞に推している人までいましたが，あのゲーム，それほどよかったのでしょうか。確かに中盤あたりはそこそこ楽しめますが，最後が最悪です。あんなに後味の悪いエンディングのゲームは「同級生」が初めてでした。主人公に強い嫌悪感を抱き，同時に主人公＝自分ですからものすごい後ろめたさを感じたゲームでした。　木村　奨(22)兵庫県

アンケートの集計結果は今月号で明らかになっています。どうです？　木村さんの思っていたような結果でしたか。

◆4月号のSTUDIO Xで紹介した11HIT COMBOですが，フェイロンはザンギエフにのみ，めくり大キックから立ち小パンチが7発入って，烈火拳が3発入ります（ただし連射装置がないとキツイ）。また小パンチは9発入るというウワサですが，未確認です。バイソンに対しては，めくりから小パンチ9発，小ダッシュストレートが入ります。なお，SUPER COMBO未使用というのはなにかの間違いです(友人に指摘された)。さらにおまけですが，リュウ，ケンの立ち大キックはある条件を満たすとキャンセルできます。連続技には使えないけどかっこいいですよ。

江森　健太郎(19)大阪府

ある程度煮詰まってしまった感のある格闘ゲームですが，それでも新しいことが発見されるあたり奥が深いなあ。

◆先日，やっとのことでX68000版「キャメルトライ」を入手することができた。しかし，キャラセレクトってどうやるんだ？　スタートボタンとジャンプボタン(?)が一緒だからわかんね～や(アーケード版ではボタンを押しながらスタートだった……はず)。

今井　健生(23)奈良県

X68000版では，メインキーの“1”と“P”を押しながらスタート，つまり“1Pスタート”とすればOKです。

◆X68000で作りたいゲームのアイデアが大量にあるのに，なかなか技術が追いつかない。グラフィックならいくらでも描けるのだが，いかんせん時間がうまくとれない。それにしても最近のゲームは，ありきたりのものばかりでイヤ。格闘ゲームも3Dシューティングゲームも似たものばかり。もっと頭を使えよなあ。

遠山　幸男(19)岡山県

遠山さんが頭を使って作った作品をぜひ見てみたいですね。

◆新品で買ったディスプレイ(CZ-608D)が1カ月もたたないうちに壊れた。ふと横を見たら“93年1～6月期”のシールが……。でも，やるせないなあ。　小林　卓生(24)石川県

ようやく活動できて，ついついディスプレイもはりきりすぎてしまったんですよ。

◆4月号のSTUDIO Xを読んでいると，RISC CPUのマシン語がいじれないという雰囲気が濃いけど，そうでもないですよ。私は過去にX-Window上から直接だけど，PA-RISC1.1とSPARCをいじってましたが結構面白かったですよ。

斉藤　誠(26)愛知県

別にアセンブラを使えなくったって，コンピュータに密着した処理を自由にできる環境があれば問題はないんですけどね。

◆SCSI，RAMボードTS-6BS1mkIIですが，X68000用としては数少ない新製品紹介なのだから，もっとしっかり紹介してほしい。特にメモリウェイトの有無が書いてありません。できれば何MHzまでのクロックアップに耐えるかまで書いてほしい。いくらスロット不足が解消できるといえどもマシンのパフォーマンスが落ちてしまうなら不要だ。　大島　博昭(21)愛知県

この大島さんの疑問をすっきり解決してくれるレポートが，次の福知さんから届いていました。

◆TS-6BS1mkIIと60nsSIMMメモリを買って，家のX68000EXPERT(17MHz)に取りつけたのだが，17MHz，16MHz，15MHzではメモリがついてこなかった。悲しい。結局，10MHzで使うハメに……。

福知　健(23)京都府

という結果になったそうですよ，大島さん。一応，念のためいっておくと基本的に拡張スロットへ増設メモリを登載した場合，10MHz以上のマシンでは，必ずメモリアクセスをするたびにウェイトがかかります(ノーウェイトでやれないことはないらしいが，とんでもなく面倒なようです)。そして，クロックアップなどの改造を行っている場合には，動作保証外の改造を施しているのですから，それなりのリスクを覚悟しておきましょう。

◆AWESOME-X，なかなか姿を現わしませんね。ツクモ電器にデビュー(?)した頃のチラシを見ると，MIDI端子ボードのほかにSIMMソケットもオプションになるはずだったのに，最近のOh!Xを見るといつのまにか消えてしまったようですね。世の中にはマーキュリーユニットとかいう，どこかで聞いたようなPCMボードもあるようですが，通信をやっていないのでよく知りません。そして，我らが瀧さんも高機能なボードを作っているそうですが，XVIな私の意見としてPCM系の端子のほかに純正互換のMIDI端子をつけてもらいたいです。残りのスロットにツクモ電器の4MバイトRAMを差せれば完璧です。がんばってください。　久保田　智久(19)群馬県

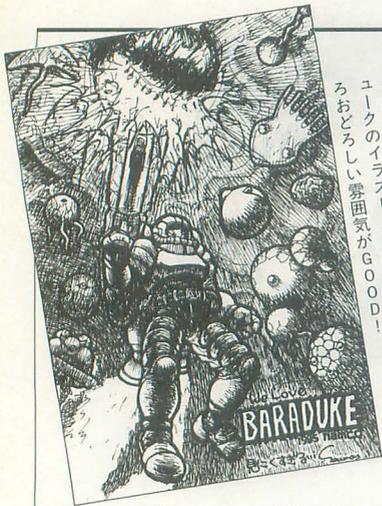
◆5月号の瀧さんのSRAMドライブ。期待しています！　氏家　隆宏(25)北海道

残念ながらSRAMドライブはちょっとだけ延期。精力的に動き続けてはいますが，ちょっとだけオーバーワーク気味かな。瀧氏の単位の無事を祈りつつ，励ましのハガキを送ってあげましょう。また，なにか要望がありましたらアンケートハガキでどしどしお寄せくださいね。

◆アルプス電子のグラインドポイントを買った。現在，純正マウスの基板を，バージニアスリムライトの箱に詰めた自作アダプタを介してつながっている。トン，でクリック，トン，ツーでドラッグ。もじもじと指を動かすとそれに合わせてマウスカーソルが動く。なかなか新鮮な感覚。しかし，細かい作業にはまるで向いていない(笑)。　清水　智明(27)神奈川県

◆さてさて，私のオーディオCDプレイヤーもついに他界してしまった。で，この際CD-ROMドライブでも，と思ったんだけど2倍速かつX68000でもOKってやつがもう売っていなかったりする。世の中完全に4倍速なんだってさ。なんてぇこ

◀岸　健司　千葉県
GAME OF THE YEARに間に合わなかったけど，試験管を楽しそうに投げつける飛鳥が気に入ったので，2カ月遅れでも掲載しちゃう。

◀今井　健生　奈良県
香織もいいけど，個人的には悠のほうが好き。ということで今度は悠のイラストをリクエストしたいな。よろしくね。

◀澤田　友伸　兵庫県
ゲームアンソロジーシリーズvol.13のバラデュークのイラスト。ちょっと見にくいけど，おどろおどろしい雰囲気がGOOD！

ったい。　西山　新志(23)福岡県

◆1Gバイトのハードディスクを2台買った。あまりの速さと容量に感動した。　宝福　公司(26)北海道

◆QuantumのEMP1080S(新品)を，38,800円(税別)で買ってしまった。ラッキ～。こうなると同期転送のSCSIボードがほしいですね。　嶋田　裕文(23)大阪府

◆とうとうX68000 CompactXVIちゃんを8Mバイトにしました。それにしてもあんな小さなボードが6万円近くするなんて。でも嬉しいです。ついでにビデオボードも買いました。某ツクモの店員に使用できることを念を押して確認したのに，やっぱり使えませんでした。結局，ビデオボードは友人所有のX68000 ACEについています。　荘司　康子(20)東京都

◆やっとお金がたまったので，メモリを増設することができます。いままでのメインメモリ2Mバイトからやっと解放されます。もうちょっとお金があれば，SX-WINDOW ver.3.1が買えるのに……。やっぱりぜいたくかな。どうせならXellent30sに72ピンのSIMMメモリをひとつでも載せられたら最高なのになあ。ついでに安く。　中村　慶彦(16)山口県

春になったからかどうかはわかりませんが，周辺機器をパワーアップした，というハガキがこのほかにも結構目につきました。うん，みんな元気にX68000を使っているんだなあ。よしよし。

◆最近記事のなかに新機種(X68000ではない)のことをほのめかす文があるように感じる。そろそろ出るのだろうか。　奈良　雅雄(21)神奈川県

出てもらわなくては困るんですけどね。

◆次世代X68000……あるんですか？　木下　達也(23)愛媛県

ここまで待たされると実にさまざまなウワサ，憶測が飛び交います。なにはともあれ，メーカーのやる気にかけるしかありません。

◆がんばれ編集部。NEW Xの発表は近いとの噂。X魂の火をここで消してはさびしすぎる。それにしても新しいソフトが出なくても，新しいハードが出なくても成り立つXユーザーって，つわものの集まり？　それともOh!Xのおかげかな？　坪井　秀次(23)静岡県

Oh!Xが成り立っているのは，ちゃんとX68000を使ってくれる読者の皆さんがいるからです。感謝，感謝。

◆ある雑誌にOh!Xが廃刊になるとか書いてあったが本当かなあ。　久保　正文(25)香川県

それはウソです。

◆「X68000ソフト情報(大阪日本橋編)」最近，X68000のソフトを置いてある店が，また減りつつありますよね。大阪日本橋でも置いてあるのはJ&Pテクノランドとスタンバイだけになったみたいです。あのソフマップでもX68000のソフトは「ディグダグ/ディグダグII」だけですし(ニノミヤは行かないので知りません)。中古ソフトならスタンバイとソフマップに結構置いてあります。　大久保　貴司(22)大阪府

ということで日本橋近辺のX68000ユーザーは参考にしてください。また，このほかの地域の情報もお待ちしてますよ。

◆まさか勝手にGAME OF THE YEARに載るとは……。自分の書いた賞が目に入ったとき，じわっと冷や汗が出てしまった。北畠　駿(13)滋賀県

ふっふっふ。びびっとるびびっとる。これを見たらどんなリアクションを見せてくれるのかちょっとだけ楽しみだなあ。

◆PC-6601を買ってもらってから，12年間，ひとりとしてパソコンを使っている人に出会わなかった。そんなものなのでしょうか，やっぱり。本当は「電子工学科」にも合格していたけど，せっかく「薬学部」に合格したのだから，と薬科大学に通うことになった。X68000 CompactXVIがほしい。でも，ひとりでなにができるのだろう。　石川　真(18)千葉県

石井さんは，5月号のアマチュアCGAコンテストの入賞作品をご覧になりました？　たったひとりでもあのような見事な作品を作り上げられるのですから，要はやる気です。それに仲間を探したいのならパソコン通信を始めてみては？

◆1984年11月号で初めてOh!MZ(現Oh!X)を買った。それから10年後。1995年4月号22ページで初めて自分の名前が載る。ヤッター！　堀川　満平(30)大阪府

おめでとうございます！　今度は常連目指してがんばろう。

◆Z-MUSICシステムはもう売っていないのですか。また，ソフトバンクでは通販を行っていないのですか。そのへんがちょっと知りたいです。それから編集部のなかの人でサリン事件に巻き込まれた人はいますか？　あの事件は最悪ですね。　加藤　幸(20)北海道

通販自体は行っていません。入手するにはソフトバンクの出版事業部販売局に問い合わせて在庫確認のうえ，本屋さんに注文して待つ。もしくは期日は未定ですがZ-MUSIC ver.3.0の発売を待つ。というような方法が考えられます。

◆このメッセージが届く頃には，サリンのことか仮谷さんのことが解決している(かもしれない)と思います。しかし，怖いですよねサリンって。たったの2.5gくらいで小錦が……(たとえですよたとえ)つまり，0.7gで自分自身が(なんで自分なの)ということは，人間ひとり80kgだとして世界人口が60億人とすると，4,800トンのサリンがあれば人類は滅びる……。　石井　義尚(19)神奈川県

残念ながらハガキが届いた時点では，状況はさらに混沌としています。6月号が発売されるときにはすべてが解決しているといいなあ。

◆友人がフライングをして20歳にして就職&結婚&双子のパパになりました。しかもその奥さんは私のよく知る後輩だったので，まさに人生いろいろです。T君夫妻，これからもお幸せに(このセリフをいってみたかった)。　片倉　純也(20)宮城県

いってみたかった，という気持ちはわかりますが，はたしてOh!Xに送ったところで本人たちに伝わるかどうか。それともT君はOh!Xの読者なのかな。

◆ほっとひと息。女子浪人生読者にならずにすみました。女子大生読者なら，きっと女子高生読者よりお仲間が多いですよね。　松尾　絢(18)東京都

……たぶん。

▲藤澤　篤　奈良県
どうせだったら真サムライスピリッツだけでなくギャラクシーファイト，餓狼伝説3も一緒に移植希望してしまいましょう。

▲澤田　友伸　兵庫県
とにかくいうだけはただ。どれか発表になるといいなと，いっているそばからバラデュークは発売決定！　よかったね。

◆今日，私の大学の研究室で爆発事故が起こった。これで二度目だ。ちなみに現場近くでは異臭がした。　　　高橋　健滋(19)東京都

今月のデンジャラス度ナンバー1のハガキ。高橋さん，巻き込まれないように気をつけてくださいね。

◆日本人は“熱しやすく冷めやすい”のではなく，マスコミに踊らされやすいのではないか。2年前の異常なJリーグ人気は明らかにマスコミの影響だし，いまは人気が落ちたと始まる前からいっている。まあ，もとから好きだった人，本当に好きになった人だけが残るのだろうから，チケットを取るぶんにはいいのだけれど。テレビ中継にも影響があるから……やっぱり困る。

三浦　貴至(23)埼玉県

テレビ中継がなければ競技場まで足を運べばいいだけのこと。やっぱりライブが……

「あれえ，編集長～どこ行くんですか？」

「三沢！」……ね。

◆中国は雲南省大理へ行ってきました。上海から汽車で3泊4日のあと，バスで11時間。ようやく大理に着くや，長旅の疲れのせいか貧血で倒れてしまいました。が，万年雪をいただく山々や三塔寺の美しさに旅の疲れもどこへやら。決して楽とはいえない旅行でしたが，それだけの価値はありました。　　　松永　貴輝(24)大阪府

壮大な自然のなかにいると，なんだか別世界にいる気分になりますよね。

▲小川　伸輔　宮城県
もちろんXellent30sでパワーアップするためにばらすんですよね。しかし，そうなったら彼女の人格(?)はどうなるのだろうか。

▲岡村　直也　兵庫県
二人の仲はいきなり進展を見せたのか，それともこれは学君の妄想なのか。この後の展開が非常に気になるが……え～続かないの？

◆4月号で試験前の幸福量のことを載せていただきましたが，あのあと，ヤマは当たりまくり，某誌の懸賞でPSを当て，イベントはハガキを出せばほとんど招待状が届く，というような状態(もちろん，ハズレもあるが当選率が違う)。もう，幸福量なんて金輪際信じません。信じないもん。　　　西川　和範(20)東京都

よかったですねと素直に喜んであげたいところですが，なんとなく悔しい。

◆NHKスペシャル「生命」の最終回を見て爆笑してしまった。そこには「恐竜都市-1997」にそっくりなカットが多数あった。CG業界が先細りというのはあながち間違いではないのかも。

音羽　進(20)宮城県

人材不足はどこでも一緒。てなことですかね。

◆ついに4月から就職です。職場も福岡市博多区東公園に決まり，いよいよどきどきしてきました。さまざまな不安のなか，4月号を読んでいると岩瀬さんの元気なイラストが……。私もこんな輝いた目でがんばろうと思います。ヨシ！　　　堂領　輝昌(21)宮崎県

つらいことがあっても初心を忘れないようにがんばろうね。

ぼくらの掲示板

●掲載ご希望の方は，官製ハガキに項目(売る・買う・氏名・年齢・連絡方法……)を明記してお申し込みください。
●ソフトの売買，交換については，いっさい掲載できません。
●取り引きについては当編集部では責任を負いかねます。
●応募者多数の場合，掲載できない場合もあります。
●紹介を希望されるサークルは必ず会誌の見本を送ってください。

仲間

★当サークル「AGLAIA」では機種非限定のディスク会報を発行しています。今回，会報の活性化のため，非会員からの投稿も募りたいと思います。だべり，自作プログラム，質問，売買，そのほかなんでもかまいません。活動に参加する意志がある人はMS-DOSの2HD(8セクタタイプ)フォーマットのディスク(3.5インチ，5インチ可)に入れてお送りください。その際，住所を書いたラベル，返送用切手を貼った返信用封筒を同封していただければ，次回に発行する投稿された記事を載せた会報をお送りします。〒811-42　福岡県遠賀郡岡垣町戸切794-3　筑紫　高宏

売ります

★シャープのSCSIボード「CZ-6BS1」を14,000円で売ります。箱，マニュアル，付属品ありで完動美品です。連絡は往復ハガキでお願いします。〒573　大阪府枚方市牧野北町10-34-107　橋本　智也(19)

★ローランドのMIDIモジュール「CM-64」，シャープのMIDIボード「CZ-6BM1」をセットにして50,000円で売ります。説明書，箱はあります。連絡は往復ハガキでお願いします。〒243　神奈川県厚木市長谷260-16　宏安　清人(41)

★シャープの熱転写プリンタ「CZ-8PC3」を12,000円で売ります(送料当方負担)。箱，説明書，付属品あります。連絡は往復ハガキでお願いします。〒312　茨城県ひたちなか市津田2634-7　友部　和彦(23)

★東京システムリサーチのX68000CompactXVI専用増設SIMMメモリボード＋4Mバイト増設SIMMメモリ(72ピン，60ns，RED ZONEでも使用可能)を25,000円で売ります。箱などすべてあります。連絡は往復ハガキでお願いします。〒223　神奈川県横浜市新吉田町2600　日本工英　網島寮305　丸山　英一(21)

★I・Oデータの4Mバイトメモリボード「PIO-6BE4」を15,000円で売ります。箱，説明書ありの完動品です。連絡は往復ハガキでお願いします。なお，送料は別です。〒228　神奈川県相模原市東林間4-4-1　ハーフテリア大野305　設楽　博幸(26)

★I・Oデータの4Mバイトメモリボード「PIO-6BE4」を35,000円，アイテックの80Mバイトハードディスク(40＋40のSASI)「ITX-680」を25,000円で売ります(早いもの勝ち)。送料は当方が負担します。〒167　東京都杉並区上井草1-30-15-401　秋山　和徳

買います

★ツクモ電機のMIDIボード「TS-6GM1」を10,000円で買います。連絡は官製ハガキでお願いします。〒040　北海道函館市高盛町14-6　船越　直弥(22)

編集室から from E・D・I・T・O・R

DRIVE ON

このコーナーでは，本誌年間モニタの方々のご意見を紹介しています。今月は4月号の内容に関するレポートです。

●特集を読んで思ったのですが，いまさらながら，X68000はアクション系のゲームが多いマシンですね。コンシューマ機のようです。しかし，注目すべきは「完成度の高いオリジナルがあったこと」ではないでしょうか？「好きなX68000のゲームをひとつだけいえ」といわれるとすごく悩みますよ，ホント。

そこで「GAME OF THE YEAR」です。「X68000でしかできないオリジナルゲームが大賞になってほしい」と思っていたので，「ジオグラフシール」の受賞は「うむ，納得である」って感じです。新型ゲーム機でなければ開発できないようなポリゴンゲームをX68000で実現してくれたわけですから。

あと，移植ゲームもなかなかの完成度でした。特に「餓狼伝説SPECIAL」は壮絶でしたね。「スーパーストリートファイターII」もすごかったけど，「要メモリ4Mバイト，推奨クロック16MHz」といった条件つきでした。それを，一般的な環境（メモリ2Mバイト，クロック10MHz）で，あの移植度を実現したのですから，いやはやまいりました。

これからも遊ぶからには完成度の高いゲームをプレイしたいものですね。ネッ「○I○W……」クン！

中矢　史朗(24)　X68030，X68000 ACE-HD，PC-386P　愛媛県

●今回，紹介された2つのハードについていわせていただきます。

まず，「TS-6BSImkII」ですが，「ついに出たSCSIメモリボード」といわれても，「いまごろなにをいっているのですか。両方とももっているし，すでに拡張スロットはいっぱいです」というユーザーは多いのではないでしょうか。という私もそのひとりなのです。SCSIメモリボードとすでに所有しているボード2枚を無料で交換してもらえるなら話は別ですが。

とはいえ，どちらももっていないユーザーにとっては非常に魅力的な製品だと思います。仮に私がどちらももっていないとしたら，ショップに走っていることでしょう。しかし，基本的にはSCSIボードなのでメモリを別に購入しなくてはなりません。すでにどちらかのボードをもっている人にはメリットがないように思います。そのあたりを考えたサービスがほしいところですね

次に「DōGA CGアニメーション講座」で紹介された「オーバーレイユニット」についてです。私の場合，カラーイメージユニットを所有しているため，オーバーレイユニットを購入したときのメリットはクロマキー合成ができるということになります。クロマキー合成を実現する商品は非常に珍しいので，個人的には興味をもちました。ただ，これを使うためには「スキャンコンバータ」が必要というのが気になる点です。スキャンコンバータの機能を含めての価格であれば，もっと普及すると思います。

壁谷　善嗣(35)　X68000 EXPERT，PC-9821As，PC-9801NS/E　宮城県

●「XDTP SX-68K」に関する連載はなかなかよかったと思います。実際に使用するうえで，つまずきそうな点を写真入りで解説しているので，XDTPを使っている人にとっては役にたったのではないでしょうか。

一方，この連載では，XDTPとSX-WINDOWの問題点が浮かび上がったわけですが，シャープさんにはぜひとも改善に努めてもらいたいものですね。

矢野　啓介(21)　X68000 XVI，MZ-2500　北海道

●温故知新，いい言葉だ。なにごとも先のものを追うばかりが能ではない。今回の特集記事に触発されて思わず「ギャラガ88」と「ナイアス」をクリアしてしまった。まったく罪なゲームだぜ。あまりに理不尽な殺され方に血管が切れそうになったことが，いまではいい思い出だ。これからも「ク○ゲー」とわかりつつも思わずプレイしてしまうようなゲームを作ってほしいぞ。

小林　佳徳(21)　X68000 XVI　新潟県

●「ピコピコエンジン活用講座」でリスト1の126～145行を見てがっかりです。たったこれだけのことをするのに，こんな行数を割かなければいけないなんて，やっぱりBASICじゃありません。こんなのは2，3行でできるべきことです。こういう上品でダラダラしたリストは人前に出すものじゃありません。これではSX-BASICもC言語などと同じです。C言語を使うレベルのユーザーを対象にしています。それなら，C言語を使えばいいじゃないですか！　無駄に言語を増やしているだけです。BASICというのは，キーを1文字打つのも待ちきれない，1秒でも早くゲームを作りたい，遊びたい，というジョイフルな人間のための言語なんです。拡張性なんて上品ぶったものは（少ししか）いらない。誰かHu-BASICのようなコテコテで強力なBASICを理解してくれる人はいないのですかー！

「X-BASICってC言語みたいでカッコイイ」という勘違いはいいかげんやめましょう。

鈴木　朝夫(20)　X68000，MZ-1500，X1turbo Z，PC-9801RA，PC-88VA2，PC-6601SR，FM77AV40SX，ZX-81，MSXturboR　神奈川県

●「知能機械概論」が印象に残りました。「囚人のジレンマ」のゲームは以前から知っていて，「しっぺ返し戦略」も知っていました。ところが，「パブロフ戦略」は初めて知りました。世の中は確実に進歩しているのですね。こういう結果を見るとほか（たとえば「R.C.ロボットコンストラクション」とかトランプゲームとか）に転用したくなりますが，ゲームの形式や情報量が異なると，どうしていいやら……。それに，ここに出てくる戦略は「勝つ」よりも「負けない」戦略だしなぁ。

弦元　達也(24)　X68000ACE　香川県

ごめんなさいのコーナー

5月号　フォント＆ロゴデザインツール
書家万流 SX-68K

P.75　1段目の24行にある「いちいち1文字ずつ指定していくのは現実的ではない」とあるのは間違いで，選択文字すべてを一括して変更することができました。関係者および，読者の方々に大変ご迷惑をおかけしました。お詫びいたします。

バグに関するお問い合わせは
☎03(5642)8182(直通)
月～金曜日16：00～18：00

お問い合わせは原則として，本誌のバグ情報のみに限らせていただきます。入力法，操作法などはマニュアルをよくお読みください。

また，よくアドベンチャーゲームの解答を求めるお電話をいただきますが，本誌ではいっさいお答えできません。ご了承ください。

夢の自分だけのシステム

▶「SX-WINDOW ver.3.1」は現在ほかにあるウィンドウシステムと比べても，なかなかよいシステムではないかと思います。もちろん，SX-WINDOWなんてなくとも，Human68k上での作業のみで十分だという人もいるでしょう。

そんな人はさておき，なかなかに優れたSX-WINDOWですが，不満のある人もいるでしょうし，いろいろな問題点もあります。しかし，不満ばかりいっていてもしかたありません。それに，その問題点の一部が改善されれば，ずいぶん使いやすくなることもあります。

これまでに，Oh!X誌上でも，SX-WINDOW上で作業を快適にするために，いろいろなプログラムを紹介してきました。メガディスプレイ計画なんかがそうですね。

そして，まだSX-WINDOWには改善の余地が残されています。皆さんも，自分が快適に作業できる環境を構築してみませんか？

もちろん，この精神はHuman68k上で作業をしている人にも当てはまりますのでがんばってください。

▶「言わせてくれなくちゃだワ」に寄せられた毎年恒例のアンケートの分析はいかがだったでしょうか？　年とともに皆さんの環境も着実に向上しているようです。ただし，いくつかのアンケートに目を通すと，周辺機器やメモリ容量などが充実している人とそうでない人の差が大きかったような気がします。いまや，マイコンの味が失われつつあるパソコンですが，その最後の砦ともいえるX68000のために，今月の特別企画を参考にしてハードウェア環境も充実させてはいかがでしょうか？

▶これを書いているいまの段階では，期待の高いDSPボードは編集部に届いていません。届き次第，レポートしたいと思いますので，楽しみにお待ちください。また，松下電器が開発したPDのレポートも予定しています。

▶「X68000マシン語プログラミング」「石の言葉，言葉の夢」は今月も著者多忙のため，残念ながらお休みです。

投稿応募要領

●原稿には，住所・氏名・年齢・職業・連絡先電話番号・機種・使用言語・必要な周辺機器・マイコン歴を明記してください。

●プログラムを投稿される方は，詳しい内容の説明，利用法，できればフローチャート，変数表，メモリマップ（マシン語の場合）に，参考文献を明記し，プログラムをセーブしたフロッピーディスクを添えてお送りください。また，掲載にあたっては，編集上の都合により加筆修正させていただくことがありますのでご了承ください。

●ハードの製作などを投稿される方は，詳しい内容の説明のほかに回路図，部品表，できれば実体配線図も添えてください。編集室で検討のうえ，製作したハードが必要な場合はご連絡いたします。

●投稿者のモラルとして，他誌との二重投稿，他機種用プログラムを単に移植したものは固くお断りいたします。

あて先

〒103 東京都中央区日本橋浜町3-42-3

ソフトバンク出版部

Oh!X「テーマ名」係

S H I F T · B R E A K

▶ある日，車に乗っていると，ヒッチコックのおじさんの車に，おかまを掘られた。自分のお客さんの病院で看てもらうと，開口一番「診断書何カ月で切ろうか？」。怪訝な顔をしていると「後遺症の恐れありなんて書くと，いっぱいもらえるよ」とダメ押し。結局，金ではなくて徹底的な修理で手をうった。ちなみに，彼の病院は震災で半壊した。合掌。　(文)

▶毎度買い物の話でスマン(しかもＡＴ)。(で)氏からi486DX4-100を譲ってもらったのだが，どうやら壊れていたようだ。残念。それとも私のマシンが悪かったのか？　話は変わるが，CPUアクセラレータが完成しないまま海外に逃亡してしまった石上氏に代わって，瀧氏がグラフィックアクセラレータを作ってくれるそうだ。期待しよう。　(I.K)

▶EXEクラブの電卓のナンバーは，X68030になってから初期化されてたりする。電卓の文字もX68030になっている。どういうわけか，X68030電卓は3つももってるのだ。X68000電卓は2つだったかな。思えば，いっぱいマシンを壊したなぁ。わたしゃ，本体に名前つけてやる愛情なんてないぞ。コンピュータは私の「しもべ」なのだ。　(瀧)

▶おそらくバグはいちばん多かった。締め切りを破ったのも一度や二度ではなかった。それでもコンピュータの核に触れていたかった。アクセラレータは満足できるところまでいけなかったが，SX-BASICはたいへん楽しい経験になった。OSやアセンブラを作っていた8ビット時代もよい思い出だ。このような機会を設けてくれた編集者，読者に感謝。　(石)

▶C++を始めた。Code Warriortというソソる名前の開発パッケージは，C，C++，Pascalのコンパイラが一式セットで13,000円強と，Macintoshのソフトにしては破格の安値。そして付属のクラスライブラリを使えば，数行のソースを書くだけで，素朴なテキストエディタが作れてしまうのだ。しかし，マニュアルがすべて英語なのにはまいった。　(ats)

▶個人的にリムーバブルメディアって他人とデータ交換ができてこそだと思う。だけど，いまはMO，PD，zipのどれが主流になるかわからない情勢。困ったもんだ。とりあえず，zipドライブを買おうと思うのだけど，私の買うものってみんなマイナーになるジンクスもあってちょっと怖い……。皆さんはどれにします。　(230なMOユーザーでもある(で))

▶エースドライバーとセガラリーはいまだに私の中で評価が揺れている。元ねたのレースカテゴリーは正反対だが，丁寧に走ってラップタイムを短縮することでのみ達成感を得られるという点で奇妙な一致をみる。1周を走り切ること自体がすでにチャレンジングなデイトナは奇跡的なゲームだと改めて思うのだ。　(最近仕事とゲームしかしてない A.T.)

▶実をいうと，生まれて初めて選挙というヤツに行った。東京都知事選だ。これは青島幸男がいくと，立候補者の顔ぶれを見たとき思った。これは青島幸男に入れようと。あらゆる旧い構造が変わろうとしているから，それに賛成するという意味でだ。多くの人がそう思ったのだろう。問題は，過渡期のさまざまな憂鬱をみんな耐えられるかどうかだ。　(K)

▶先日購入したHP200LXは快調に動作中。それにしても，本体と10Mバイトのフラッシュパッカーで12万円強，瀧氏が発見してきたCompactXVIの中古ならもう少し払えば2台買えた。現在，X68000を所有していない私にはグラっとくるお値段である。来月にはパソコンがもう1台増えているかもしれない？
(ファイル共有の番外編が楽しみな高)

▶先日，以前注文した話題のアストニッシュが，我が家に届いたので，さっそくガスレンジ周りを拭いてみた。……落ちない。うちのガスレンジは4,000度のバーナーであぶった汚れよりも強力なのか。じゃ，もう少し汚れの少ない電子レンジはどうだ。多少落ちるが疲れるぞ。宣伝とはエライ違いだ。ちくしょう。マイク・レビン，カネ帰せ！（笑）　(J)

▶最近↓で景気の悪い話ばかりしているのでなんだが，今月は広告の数がまた素晴らしく……いやあ，これでよくやってけるよなぁ，経営努力だよなぁとちょっと感心(こらこら)。しかし，最近の選挙って政党名を名乗るのって共産党だけになったなあ。ちょっと昔は候補者名より政党名のほうを連呼してた人も多かったのに。　(U)

▶「もしもこの秋に次世代マシンが発表されたら？」と聞かれ，そんときゃ「ありがとうX68000」でもやるか，なんて口走っちゃったものだからさあ大変。DoGAの鎌田さんの耳に入るころには「Oh!X 9月休刊説」になっていた。伝言ゲームは怖い。でも，時代の流れに逆らって，Oh!Xがいまだに存続しているのはパソコン界の謎だ(ほとんど自画自賛)。(T)

microOdyssey

「人は見かけによらぬもの」「人は上辺によらぬもの」「あの声で蜥蜴食らうか時鳥」「外面似菩薩内心如夜叉」などと，人の外見から内面は判断できないという類の故事，諺はたくさんある。

もちろんそれは正しい。その人の内面を知るにはそれなりのつき合いと時間が必要になる。だが，「見かけ」で判断して当たっていることもあるし，現在の世の中では見かけで判断されることは少なくない。今年も読者の中には就職活動をする人も多いだろう。その際に行われる面接などは，「見かけ」で判断される最たる例だ。

「見かけ」というと，つい顔のことを思い浮かべてしまう。そういえば，こんなテレビCMがあった。面接らしきシーンで「顔だけで世の中渡っていけると思ってない？」と聞かれて「思ってます。私，脱いでもすごいんです」と答える，あれだ(関東ローカルかも)。

これはエステ会社のCMだから体のほうにポイントがあるのだが，顔が与える印象について考えてみよう。

顔の記憶について，魅力的な人とそうでない人の印象度がUの字をなすという統計結果がある。つまり，魅力的な人はもちろんのこと，きらいな顔の人も普通の顔の人より覚えているのだ。そして，別の統計では，明るい，暗い，優しい，怖いをイメージさせる顔の特徴もわかっている。ということは，もって生まれた顔で面接などの有利不利が存在しているわけだ。しかし，もって生まれた顔はしようがない。でも，後天的なものなら，どうにかなるかもしれない。顔に刻まれるシワだ。シワができる原因のひとつに顔の筋肉を動かすというのがある。そう，無表情で顔の筋肉をあまり動かすことがなければ，シワがなく若々しさを保てるのだ(なんか違うような気がする)。そういえば，日本人が年齢のわりに若く見えるというが，案外，表情の少なさと相関関係があるのかもしれない。

話がそれたが，面接官は顔でその人のすべてを判断するわけではない。「目は口ほどにものをいう」という諺があるとおり，面接官が注意する重要なポイントに目の輝きがある。これ以外にも，姿勢のよさ，質問されていないときの仕草や行動などだ。つまり，「見かけ」といってもいろいろな要素がある。そしてポイントがわかっていれば，それなりに演技することも可能なはずだ。それで騙しきれるなら，それも技能のひとつなのだから，騙した罪悪感などにとらわれる必要はない。

ある雑誌で「初対面の人の顔を見て，相手の性格を見抜くことができますか？」という質問をしていた。その結果，10代〜30代では「できない」が「できる」の2倍以上であった。ところが，40代以上になると両者の答えは伯仲した。要は経験による自信の現れなのだろう。

ただ，40代以上の人でも半数が初対面では性格を見抜くことはできないと答えているわけだ。自信をもって臨めば，あなたの前に座る面接官も意外と簡単に丸め込めるかも……。もちろん，この結果は人事担当官だけが答えたものではないので，丸め込めなかったからといって怒らないように。そして，来年の春は就職する人だけでなく，皆に平和な春が訪れんことを。（高）

1995年7月号6月17日(土)発売

特集　最適化の手法を探る

・68000の基本技を見る
・Xellent30を生かしきる

新製品紹介　PDドライブ/AWESOME-X
ショウレポート　ビジネスショウ'95
全機種共通システム
S-OSねちねち入門(4)

バックナンバー常備店

東京	神保町	三省堂神田本店5F	03(3233)3312
	〃	書泉ブックマートBI	03(3294)0011
	〃	書泉グランデ5F	03(3295)0011
	秋葉原	T-ZONE 7Fブックゾーン	03(3257)2660
	八重洲	八重洲ブックセンター3F	03(3281)1811
	新宿	紀伊国屋書店本店	03(3354)0131
	高田馬場	未来堂書店	03(3209)0656
	渋谷	大盛堂書店	03(3463)0511
	池袋	旭屋書店池袋店	03(3986)0311
	八王子	くまざわ書店八王子本店	0426(25)1201
神奈川	厚木	有隣堂厚木店	0462(23)4111
	平塚	文教堂四の宮店	0463(54)2880
千葉	柏	新星堂カルチェ5	0471(64)8551
	船橋	リブロ船橋店	0474(25)0111
	〃	芳林堂書店津田沼店	0474(78)3737
	千葉	多田屋千葉セントラルプラザ店	043(224)1333
埼玉	川越	黒田書店	0492(25)3138
	川口	岩渕書店	0482(52)2190
茨城	水戸	川又書店駅前店	0292(31)0102
大阪	北区	旭屋書店本店	06(313)1191
	都島区	駸々堂京橋店	06(353)2413
京都	中京区	オーム社書店	075(221)0280
愛知	名古屋	三省堂名古屋店	052(562)0077
	〃	パソコンΣ上前津店	052(251)8334
	刈谷	三洋堂書店刈谷店	0566(24)1134
長野	飯田	平安堂飯田店	0265(24)4545
北海道	室蘭	室蘭工業大学生協	0143(44)6060

定期購読のお知らせ

Oh!Xの定期購読をご希望の方は綴じ込みの振替用紙の「申込書」欄にある『新規』『継続』のいずれかに○をつけ、必要事項を明記のうえ、郵便局で購読料をお振り込みください。その際渡される半券は領収書になっていますので、大切に保管してください。なお、すでに定期購読をご利用の方には期限終了の少し前にご通知いたします。継続希望の方は、上記と同じ要領でお申し込みください。

基本的に、定期購読に関することは販売局で一括して行っています。住所変更など問題が生じた場合は、Oh!X編集部ではなくソフトバンク販売局へお問い合わせください。

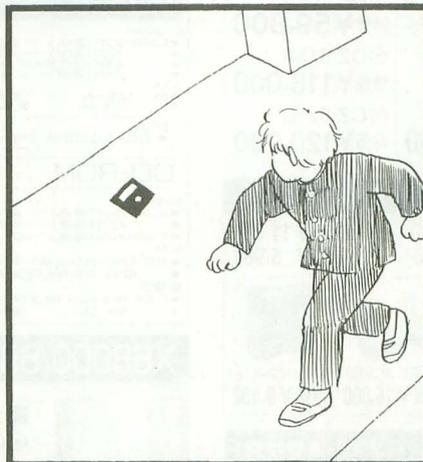
海外送付ご希望の方へ

本誌の海外発送代理店、日本IPS(株)にお申し込みください。なお、購読料金は郵送方法、地域によって異なりますので、下記宛必ずお問い合わせください。

日本IPS株式会社
〒101 東京都千代田区飯田橋3-11-6
☎03(3238)0700

Oh!X 6月号
■1995年6月1日発行 定価680円(本体660円)
■発行人 橋本五郎
■編集人 稲葉俊夫
■発売元 ソフトバンク株式会社
■出版事業部 〒103 東京都中央区日本橋浜町3-42-3
Oh!X編集部 ☎03(5642)8122
販売局 ☎03(5642)8100 FAX 03(5641)3424
広告局 ☎03(5642)8111
■印刷 凸版印刷株式会社

満開の電子ちゃん

作・え 岡村 祭

84号(4/18発送)では、BMP(WIN)、PICT(Mac)、MSXの画像ローダ、「Xellent30s」「学研辞書for SX」「書家万流」レポ、暴走大学C講座が開設!!

購読方法：定期購読、ソフトベンダーTAKERU、NIFTY-SERVEでお買い求めいただけます。
また、JCB、VISAカードもご利用になれます(金額9,000円以上の場合)。
★定期購読(送料サービス、消費税込)3ヶ月＝4,500円、6ヶ月＝9,000円、12ヶ月＝18,000円。
・現金書留：〒171 東京都豊島区長崎1-28-23 Muse西池袋2F (株)満開製作所
・郵便振替：02810-6-13298 口座名 電脳倶楽部
・JCB・VISAカード：フリーダイヤル0120-887780 または、NIFTY-SERVE GO MANKAI。
ご注文の際には、郵便番号、住所、氏名、電話番号、タイプ(5インチ・3.5インチ)、新規購読か継続購読かを必ずお知らせ下さい。新規購読の際、購読開始号のご指定のない場合は既刊の最新号よりお送りいたします。製品の性格上返品には応じられませんが、お申し出があれば定期購読を解約し残金をお返しいたします。
★TAKERUでお求めの場合、75号までは1,200円(税込)、76号以降1部1,600円(税込)です。
★お問合わせ先 TEL03-3554-9282(月～金 午前11時～午後6時)。
★バックナンバーは創刊号よりございます。★フリーダイヤルは、午前10時～午後5時。

浅井義之(奈良県)

え～、毎度お騒がせしております満開製作所でございます。御家庭にありますX68Kは眠っておりませんか? そういう方はすぐさま電脳倶楽部を御購入くださいませ。怠けているX68Kに活を入れるツール、環境をパワーアップさせる各種デバイス接続情報がもれなくついております。

そう、電脳倶楽部はこういう地道な広報活動の上に作られているため、ユーザーの生の声を聞きとり、本当に必要とされている情報を提供できる数少ない雑誌なのです。ほら、聞こえませんか? あなたの街の電脳倶楽部売りの声が…。

マイコン専門ショップ

P&A

SHARPエキスパートショップ

今が購入のチャンス!

5/17~6/17

X68030お買い得セット

(クレジット表:送料・消費税込み)

①ハードディスクセット

- CZ-500C(本体)
- 340MB(外付)ハードディスク

定価 ¥506,000

P&A超特価 **¥255,000**

12回	23,100	24回	12,100	36回	8,400
48回	6,600	60回	5,500		

②モニターセット

- CZ-500C(本体)
- CZ-608D-B(モニター)

定価 ¥492,800

P&A超特価 **¥280,000**

12回	25,400	24回	12,300	36回	9,200
48回	7,200	60回	6,000		

(⊙本体をCZ-300C(compact)に変更の場合同額になります。)

■②のモニター変更の場合
- CZ-615D(チューナ付)に変更の場合 ¥56,000
- CZ-621D(B)…………に変更の場合 ¥64,000

加算して下さい。

X68030オリジナルセット

⊙コプロ追加の場合¥10,000加算して下さい。

⊙CZ-500C
- HD(内蔵)500MB
- メモリー8MB増設(合計12MB)
- SX-WINインストール済み

特価 **¥328,000**

⊙CZ-500C
- HD(内蔵)700MB
- メモリー8M増設(合計12MB)
- SX-WINインストール済み

特価 **¥358,000**

※CZ-300C(Compact)に変更の場合同額になります。

⊙内蔵ハードディスク(30用)
- 500MB 特価 **¥64,000**
- 700MB 特価 **¥89,000**

(取り付けOK)

決算大処分セール 旧シリーズ今が買いどき!!

(送料¥2,000・消費税別)(クレジット表:送料・消費税込み)

X68000 Compact XVI

- CZ-674C-H
- CZ-608D(B)

定価 ¥392,800

P&A超特価 **¥145,000**

12回	13,200	24回	7,000	36回	4,800	48回	3,800	60回	3,100

- CZ-674C-H
- CZ-608D(B)
- CZ-6FD5

定価 ¥492,600

P&A超特価 **¥193,000**

12回	17,600	24回	9,200	36回	6,400	48回	5,000	60回	4,200

決算大処分セール 旧シリーズ今が買いどき!!

(送料¥1,000・消費税別) 単品、限定

⊙PROII-HD
- CZ-663C

(ハードディスク40MB内蔵)

P&A超特価 **¥49,800**

⊙PROII-HD 最強モデルセット
- CZ-663C
- メモリー11MB増設(合計12M)
- SCSIボード付

P&A超特価 **¥128,000**

⊙CompactXVI
- CZ-674C

P&A超特価 **¥79,800**

X68用専用ディスプレイ
- ⊙CZ-608D 特価 **¥59,800**
- ⊙CZ-615D 特価 **¥118,000**
- ⊙CZ-621D 特価 **¥120,000**

MIDIセット

- MC-6600(SNE)
- SX-68MII(システムサコム)
- MIDIケーブル

特価 **¥48,500**

- SC-55MKII(ローランド)
- SX-68MII(システムサコム)
- MIDIケーブル

特価 **¥70,800**

(SC-88に変更の場合¥17,000加算して下さい。)

単品
- MC-6600(SNE)…………特価 ¥34,800
- SC-55MKII(ローランド)・特価 ¥56,800
- SC-88(ローランド)………特価 ¥73,500
- 05R/W(KORG、加賀電子)・特価 ¥49,800

スピーカー

- MS-3000(SNE)……特価 ¥11,500
- SC-C55(AIWA)………特価 ¥ 5,980

ALTEC ACS300 特価 **¥37,000**

ALTEC ACS100 特価 **¥16,000**

SHARP CP-A5-B 特価 **¥9,400**

X68000/68030用 メモリボード

(送料¥700・消費税別)

■I/Oデータ
- SH-5BE4-8M(30用)……特価 ¥39,500
- SH-6BE1-1ME(600C用)…特価 ¥10,200
- PIO-6BE1-AE(ACE/PRO/PROII用)…特価 ¥10,200
- PIO-6BE2-2ME(拡張スロット用)特価 ¥21,000
- PIO-6BE4-4ME(〃)特価 ¥35,300

■シャープ
- CZ-5BE4(30用)……………特価 ¥39,800
- CZ-5ME4(5BE4用増設)・特価 ¥36,500
- CZ-6BE2A(XVI用)……特価 ¥38,900
- CZ-6BE2B(XVI、674C増設)特価 ¥37,500
- CZ-6BE2D(674C用)…特価 ¥20,500

モデム&FAXモデム

(送料¥1,000)

<アイワ>
- PV-BF144(ボックス型)……………特価 ¥17,000
- PV-AFV144(液晶パネル、ボックス型)……………特価 ¥26,800
- PV-PFV144(ポケット型)……………特価 ¥22,800

<オムロン>
- ME1414BII(ボックス型)………特価 ¥17,000
- MD-144XT10V(限定)………限定 ¥30,000

<マイクロコア>
- MC14400FX(W)(ボックス型)…特価 ¥23,000
- MC24FC5(W)(ポケット型)……特価 ¥20,000

MO

(送料¥1,000)

Logitec
- LMO-200(128M)……定価 ¥69,800▶特価 ¥45,800
- 340(128M)……定価 ¥79,800▶特価 ¥52,300
- 400(128/230M)定価 ¥118,000▶特価 ¥76,000
- 420(230M)……定価 ¥138,000▶特価 ¥99,000

ICM
- MO-120S-N…………定価 ¥74,800▶特価 ¥55,000
- 230S-N…………定価 ¥118,000▶特価 ¥87,000

Filo
- CS-M230PA(230MB) 定価¥148,000▶ **¥77,800**

CD-ROM

(送料¥1,000)

Logitec
- SCD-200(2倍速)……定価 ¥19,800▶特価 ¥16,800
- 420(4倍速)……定価 ¥34,800▶特価 ¥27,200
- LCD-440(4倍速)……定価 ¥39,800▶特価 ¥26,800

ICM
- CD-620S-N(4倍速)・定価 ¥34,800▶特価 ¥26,400
- 620S-NB(4倍速・I/F付)定価 ¥44,800▶特価 ¥34,000

緑電子
- CXA-660-98(4.4倍速)定価 ¥39,800▶特価 ¥33,200
- 660-5L(〃)定価 ¥49,800▶特価 ¥44,200

東京システムリサーチ製(X SIMM)

(送料¥700・消費税別)

(X SIMM VI)

⊙XVIシリーズ専用SIMM増設式メモリボード
- X SIMM VI(634C用)…定価 ¥16,500➡特価 ¥13,000
- X SIMM VIc(674C用)・定価 ¥16,500➡特価 ¥13,000

⊙増設SIMMメモリ(72PIN)
- 4MB(70ns)……………………特価 ¥11,800
- 8MB(70ns)……………………特価 ¥27,800
- 4MB(60ns、24MHz以上用)………特価 ¥16,500
- 8MB(60ns、24MHz以上用)………特価 ¥28,000

- 6MB(60ns、メーカー純正品)……特価 ¥27,800

(X SIMM 10) ⊙SIMM増設式メモリボード
- X SIMM 10………定価 ¥18,000➡特価 ¥15,700

⊙増設SIMMメモリ
- 1MB×2……特価 ¥ 9,000
- 4MB×2……特価 ¥30,000

- 10MB例 X SIMM10+1MB×2+4MB×2‥ ¥54,700

X68000/68030専用ハードディスク

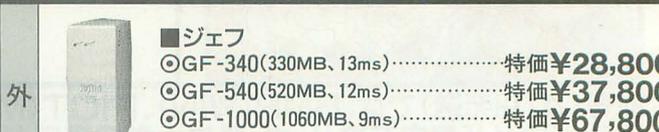
(送料¥1,000・消費税別)

外付

■ジェフ
- ⊙GF-340(330MB、13ms)……………特価 ¥28,800
- ⊙GF-540(520MB、12ms)……………特価 ¥37,800
- ⊙GF-1000(1060MB、9ms)……………特価 ¥67,800

■ロジテック
- ⊙SHD-B340AU(340MB、12ms)……特価 ¥30,800
- ⊙SHD-B540U(540MB、10.5ms)……特価 ¥42,800
- ⊙SHD-B1000U(1000MB)……………特価 ¥75,000

■富士通
- ⊙HD-M520(520MB、12ms)……………特価 ¥35,800

内蔵

■CZ-500C/300C専用
- ⊙CZ-5H08(80MB/23ms)……………定価 ¥ 98,000▶特価 ¥71,800
- ⊙CZ-5H16(160MB/18ms)……………定価 ¥135,000▶特価 ¥99,500

注目!!夏のボーナス一括払い手数料(金利)無料(平成7年6月末/7月末/8月末のいずれかをご指定下さい。)

●価格は変動します。ご注文の際は必ずお電話で価格と在庫をご確認下さい。●本広告に掲載の商品には送料及び消費税は含まれておりません。

セクシーでパワフルな女子プロを制覇しろ！

18禁版

カードバトルにプロレスを融合させた、「レッスルエンジェルス」シリーズ。いよいよ最大のヒット作「レッスルエンジェルススペシャル」が登場です。さまざまなイベントの選択によって運命が変わる、マルチシナリオ・マルチエンディング。プロレス技数、カテゴリーが増加して、レスラーの個性もパワーアップ。そして、「恐怖の水着はぎデスマッチ」もパワーアップして復活！18禁だから、そのセクシー度はもうケタ違い！待望のX68000移植完成！明日のトップイベンターを目指すのだ！

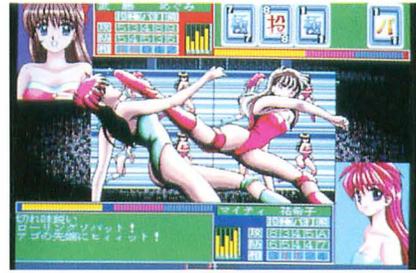
機能アップ！

- ●オリジナルオープニングを収録
- ●画面のレイアウトを変更
- ●エキジビションモードグラフィック描き直し
- ●256色モードと16色モードを搭載
- ●サウンドも明るめに変更
- ●AD-PCMによる効果音
- ●ディスクアクセスを最少に抑える設計

このソフトは、全国のパソコンショップで、パッケージ版で販売いたします。TAKERUでは販売致しません。TAKERU事務局では通信販売はいたしませんので、
悪しからずご了承下さい。

対応機種：X68000/X68030
要メモリ2Mバイト
（ハードディスク対応）

制作：グレイト

¥8,800（税別）

三國志

知力の極限に挑む、君主、武将、軍師の膨大なデータ。小説よりリアルと、名作の誉れ高い中国統一ゲーム。この歴史的な傑作シリーズはどのようにして始まったのか?SLGファンなら絶対に見逃せない!!

制作/光栄
対応機種/X68000（30不可） **¥5,200**

太閤立志伝

裸一貫の足軽頭から身を興し、関白にまで登り詰めた男・木下藤吉郎(豊臣秀吉)。草履を温めたエピソード・奇跡の墨俣一夜城など、数々の逸話を持つ男の一生を再現する、リコエイションゲームの傑作です。

制作/光栄
対応機種/X68000（30不可） **¥3,400**

ファランクス

デカキャラ・派手め演出の横スクロールオアワーシューティング。拡大・回転・縮小・多関節・半透明・ラスタースクロール・MIDIと、各種要素がいっぱい詰まってます。

制作／ズーム
対応機種／X68000（30不可） **¥2,500**

三國志 II

登場人物350余名、最大11人まで同時プレイ可能、6編のマルチシナリオ方式、埋状の毒・駈虎呑狼等のユニークな計略要素導入、さらに深みを増した外交・HEX戦など、まさに名作!カシオペアの向谷　実のBGMも話題に。

制作/光栄
対応機種/X68000（30不可） **¥4,900**

蒼き狼と白き牝鹿　元朝秘史

光栄歴史三部作の一角を成す、草原の英雄チンギス・ハーン。稀代のスケールと空前絶後の迫力で、一代帝国を築き上げた男の豪快な一生を見事に再現!熱いシミュレーションの傑作です。

制作/光栄
対応機種/X68000（30不可） **¥3,400**

A列車で行こうII

かの「A列車」シリーズの第2弾。パズル的要素がアツクなる!鉄道会社社長の立場で、線路の敷設・撤去を行い、ワールドワイドにマップを発展させていこう。

制作／アートディンク
対応機種／X68000（30不可） **¥3,800**

大航海時代

リコエイションゲームシリーズの傑作。毎回違った展開が楽しめるイベントジェネレーティングシステム。帆船の特徴が活かされたHEX戦。失われたロマンを求めて、冒険者たちの航海の旅が始まる。

制作/光栄
対応機種/X68000（30可） **¥3,400**

ロイヤルブラッド

新シリーズ「イマジネイションゲーム」のデビュー作。イシュメリアという架空の島国を舞台にした、幻想世界のシミュレーションゲームだ。あなたは独立貴族のひとりとなり、領主達が持っている6つの宝石を集め、イシュメリアの新王となれ!

制作/光栄
対応機種/X68000（30可） **¥2,700**

A III（A列車で行こう3）

さらにワイドに、さらに完成度の増した、世界レベルヒットの第3弾。世にA.IIIブームを巻き起こしたことで、記憶に新しい超有名作、ついに文庫に登場！

制作／アートディンク
対応機種／X68000（30可） **¥3,800**

維新の嵐

坂本龍馬が、西郷隆盛が、吉田松蔭が日本を憂い、改革を目指して奮い立つ!幕末の志士の個性を際だたせる緻密なパラメータ。出会いの楽しさ、駆け引きを楽しむ新システム。強力な機能で、維新を操れ!

制作/光栄
対応機種/X68000（30不可） **¥3,400**

ヨーロッパ戦線

戦乱のヨーロッパ。砂塵の彼方から迫り来る黒い車体は、敵か味方か?次々に飛び込んでくる情報、時事刻々と変わる戦局。多彩な兵器やユニット、人間的要素を重視した各種パラメータ。WWIIシリーズ第2弾。勝利の旗を手に入れろ!

制作/光栄
対応機種/X68000（30可） **¥4,500**

栄冠は君に

高校野球シミュレーションシリーズの、記念すべき第1作。全国制覇を達成するには、3990校の頂点に立たなければならない。感動の優勝セレモニーを、果たして見ることが出来るか！？

制作／アートディンク
対応機種／X68000 **¥3,800**

信長の野望　戦国群雄伝

400余名の群雄が割拠する下剋上の乱世。配下の羽柴秀吉、柴田勝家を個性豊かな武将たちを、思いのままに操って、戦雲たなびく戦場へ、天下分け目の決戦に臨む！光栄の代表作「信長の野望」シリーズの傑作！

制作/光栄
対応機種/X68000（30可） **¥3,400**

大戦略 III '90

90年代にふさわしくパワーアップされた「大戦略」シリーズ。戦略思考ルーチン、ゲームスピード、コマンド体系、リアルタイムオペレーションなど大幅革新された作品です。

制作／システムソフト
対応機種／ X68000 **¥2,500**

ルーンワース「黒衣の貴公子」

ハイドライドシリーズに続く、新ARPGシリーズ第1弾。綿密に構築された世界「ルーンワース」を舞台に、極めて自由度の高いゲームシステムの中で、興奮の冒険が始まります。

制作／T&Eソフト
対応機種／X68000 **¥700**

伊忍道　打倒信長

1つのゲームでSLG とRPG、2つのジャンルが楽しめるリコエイションゲームの第3弾。特にRPGの要素が濃い、異色傑作だ!意志を持ったキャラクターが目的に向かって行動を展開。敵を倒して腕を上げ、技を磨いて信長を倒せ!

制作/光栄
対応機種/X68000（30不可） **¥3,400**

ジェノサイド 2

あのズームのゲームがついに名作文庫に登場！特大キャラとハデハデな演出で、68ユーザーのどぎもを抜いた名作アクションゲームだ。MIDIにも対応しているぞ。

制作/ズーム
対応機種/X68000（30不可） **¥2,500**

イース III（ワンダラーズフロムイース）

よりアクション性を増した、これまた、大人気を博したアクション・ロールプレイング。アドルの最後の冒険物語でした。攻撃方法もいっそう多彩になって、時間を感じさせない逸品です。

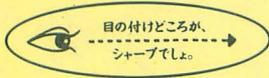
制作／日本ファルコム
対応機種／X68000（30不可） **¥2,000**

パソコンソフト
自動販売機
TAKERU

TAKERU事務局
〒467 名古屋市瑞穂区苗代町2番1号
プラザー技術開発センタービル2F
TEL(052)824-2493（受付時間：月～金 13:00～18:00）

営業所
東京営業所
(03) 5443-4967
大阪営業所
(06) 258-3024

通信販売 1994年4月1日より、送料／手数料が有料になりました。

ソフト名、機種名、メディアのサイズ、住所、氏名、電話番号を明記の上TAKERU事務局まで現金書留でお申し込みください。送料／手数料は、1回のお申し込み総金額が5,000円以上の方は無料。4,900円までの方は500円をいただきます。4,900円までの方は現金500円をプラスしてお申し込みください。誠に勝手ながら、皆様のご理解とご協力の程、お願い申し上げます。

感性を光らせる。

さまざまなフィールドで、研ぎ澄まされた感性に応える潜在能力の実証

X68の潜在能力は、まさに時代とともに証明されつつあります。
開発当初より、現在のマルチメディア環境を想定していた事実。
グラフィック能力はもちろん、ADPCM対応、オリジナルウィンドウシステム、
X68にとってこれらは、数年前のスペックなのです。
パソコンの存在そのものを革新した「創造性」、マインドを喚起する「こだわり」、
いま、先見のユーザーに支えられたX68は
そのコンセプトの開花を得て、多彩なフィールドへと飛翔します。

Workbench
WSとしての楽しみ

たとえば、リアルタイム・マルチタスク・オペレーティング・システムOS/9。
X68030の能力を最大限に引き出す
UNIXライクな操作性と洗練された機能。
X-WINDOWや動画ツールのサポートで
さらに深い楽しみが…。

※OS/9はマイクロウェア・システムズ(株)の登録商標です。
※UNIXは、X/Openカンパニーリミテッドが独占的にライセンスする米国および他の国における登録商標です。

Create
創造するよろこび

SX-WINDOW開発支援ツールが
創造力を刺激する。
ソフト開発に必要なツールや
サンプルプログラムを多彩にバンドル、
ウィンドウ上で効率よく作業でき、
初めてプログラムに挑む人への
やさしい配慮が、創造するよろこびを
さらに高めてくれるでしょう。

Ammusement
遊びへのこだわり

X68の能力の高さを端的に示す
アミューズメントフィールド。
マインドをきわめたゲームフリークの
熱い期待に応える。
画像の美しさが感性を刺激する、
さらにパワーアップされた
「スーパーストリートファイターII」なら、
キミのこだわり度は今、全開！

© CAPCOM ALL RIGHTS RESERVED

X68030 Compact　X68030　X68000 XVI Compact

X68030 / X68000
32bit PERSONAL WORKSTATION / PERSONAL WORKSTATION ·XVI

X68030 [本体+キーボード+マウス・トラックボール]
130㎜FD(5.25型)タイプ CZ-500C-B(チタンブラック) 標準価格398,000円(税別)・〈HD内蔵〉CZ-510C-B(チタンブラック) 標準価格488,000円(税別)

X68030 Compact [本体+キーボード+マウス]
90㎜FD(3.5型)タイプ CZ-300C-B(チタンブラック) 標準価格388,000円(税別)

X68000 XVI Compact [本体+キーボード+マウス]
90㎜FD(3.5型)タイプ CZ-674C-H(グレー) *

●ディスプレイは別売です。●消費税及び配送・設置・付帯工事費、使用済み商品の引き取り費等は、標準価格には含まれておりません。●画面はハメコミ合成です。
*〈標準価格〉表示のない商品の価格については、販売店にお問い合わせください。

■お問い合わせは… シャープ株式会社 電子機器事業本部システム機器営業部 〒545 大阪市阿倍野区長池町22番22号 ☎(06)621-1221(大代表)

T1002179060686 雑誌 02179-6